

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ
(ініціали, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Спеціалізовані комп'ютерні системи»
спеціальності 123 «Комп'ютерна інженерія»**

на тему: додаток на базі Android «Мобільний гід по університету»

Виконала:

студентка IV курсу, групи КВ-63
(шифр групи)

Климчук Ірина Олегівна
(прізвище, ім'я, по батькові) _____ (підпис)

Керівник доц. каф. СПКС, к.т.н. Клятченко Я. М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Консультант з нормоконтролю, доц.каф.СПКС, к.т.н. Клятченко Я.М.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) _____ (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) _____ (підпис)

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій РОМАНКЕВИЧ

(підпис)

(ініціали, прізвище)

«__» _____ 2020 р.

ЗАВДАННЯ

на дипломний проєкт студента

Климчук Ірина Олегівна

(прізвище, ім'я, по батькові)

1. Тема проєкту «Додаток на базі Android «Мобільний гід по університету»»

керівник проєкту Клятченко Ярослав Михайлович, к.т.н., доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 2020 р. № _____

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту див. Технічне завдання

4. Зміст пояснювальної записки:

Огляд існуючих рішень систем дистанційного моніторингу

Особливості програмної реалізації додатку на базі Android

Особливості розробленого програмного мобільного додатку

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Структурна схема платформи Xamarin

Схема розроблених сервісів

Схема взаємодії компонентів інтерфейсу

Блок схема алгоритмів розрахунку зміни координат

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «31» жовтня 2019р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	17.11.2019	
2.	Розроблення та узгодження технічного завдання	28.11.2019	
3.	Аналіз існуючих рішень	15.12.2019	
4.	Підготовка матеріалів першого розділу дипломного проєкту	30.12.2019	
5.	Розроблення програмного забезпечення	03.02.2020	
6.	Відлагодження програмного продукту	13.02.2020	
7.	Підготовка матеріалів другого розділу дипломного проєкту	20.02.2020	
8.	Розроблення апаратної частини системи	17.03.2020	
9.	Підготовка матеріалів третього розділу дипломного проєкту	30.03.2020	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	11.04.2020	
11.	Підготовка графічної частини дипломного проєкту	28.04.2020	
12.	Оформлення документації дипломного проєкту	18.05.2020	

Студент

(підпис)

Ірина КЛИМЧУК

(Ім'я та ПРІЗВИЩЕ)

Керівник проєкту

(підпис)

Ярослав КЛЯТЧЕНКО

(Ім'я та ПРІЗВИЩЕ)

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Метою даного дипломного проекту є розроблення застосунку для платформи Android, який забезпечує користувача інформацією про територію національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

У процесі дипломного проектування було розроблено мобільний застосунок. Для визначення поточного розташування користувача використано систему позиціонування GPS. Для визначення найближчого об'єкту території університету (корпусу) розроблено модуль пошуку напряму пристрою по відношенню до півночі. В якості основного компоненту модулю використовується сервіс розрахунку зміни геодезичних координат залежно від азимуту телефону зі зміщенням на певну кількість метрів.

Система складається з двох частин: блок обрахунку та інтерфейсний блок. Блок обрахунку включає в себе модуль пошуку напряму а також сервіси для зчитування даних про локацію, сервіс визначення поточної позиції, сервіс визначення поточної адреси за геодезичними координатами, сервіс зберігання даних. Інтерфейсний блок складається з трьох сторінок, кожна з яких призначена для відображення певної інформації про поточну локацію. Головна сторінка представляє собою форму, на якій відображається карта з поточним положенням користувача. Також на формі розміщена кнопка, яка дозволяє почати роботу модулю пошуку найближчого об'єкту. Також є перемикач, який дозволяє поставити на карті позначку і вивести інформацію про локацію на позначці. Друга сторінка призначена для відображення інформації про локацію. Складається з поля для назви, фотографії або декількох фотографій, зроблених у різний час а також короткої довідки про місце. Має вертикальну смугу прокрутки. Третя сторінка призначена для відображення загальної інформації про університет.

ABSTRACT

The purpose of this diploma project is to develop an application for the Android platform, which provides the user with information about the territory of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

In the process of diploma design, a mobile application was developed. A GPS positioning system is used to determine the current location of the user. To determine the nearest object of the university (building), a module for searching the direction of the device relative to the north has been developed. The main component of the module is the service of calculating the change of geodetic coordinates depending on the azimuth of the phone with an offset of a certain number of meters.

The system consists of two parts: the processing unit and the interface unit. The calculation unit includes a direct search module as well as services for reading location data, a service for determining the current position, a service for determining the current address by geodetic coordinates, a data storage service. The interface block consists of three pages, each of which is designed to display certain information about the current location. The main page is a form that displays a map with the current position of the user. There is also a button on the form that allows you to start the search module for the nearest object. At least, there is a switch that allows you to put a mark on the map and display location information about the mark. The second page is for displaying location information. It consists of a field for the name, photo or several photos taken at different times and a brief reference to the place. Has a vertical scroll bar. The third page is intended to display general information about the university.

[illegible]

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ.....	2
3. МЕТА РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до програмного продукту, що розробляється	3
5.2. Вимоги до апаратного забезпечення, що розробляється	3
5.3. Вимоги до програмного та апаратного забезпечення на етапі експлуатації	3
6. ЕТАПИ ПРОЄКТУВАННЯ.....	4

					<i>ІАЛЦ.045490.002 ТЗ</i>		
Змін	Арк.	№ докум.	Підпис	Дата			
<i>Розробив</i>		Климчук І. О.			Додаток на базі Android «Мобільний гід по університету» <i>Технічне завдання</i>		
<i>Перевірів</i>		Клятченко Я.М.					
<i>Н. контр.</i>		Клятченко Я.М.					
<i>Затв.</i>		Романкевич В.О.			КПІ ім. Ігоря Сікорського, ФПМ, КВ-63		
					Літ.	Аркуш	Аркушів
						1	4

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: «Додаток на базі Android «Мобільний гід по університету»»

Галузь застосування: створення мобільного застосунку, який дозволяє переглядати фотографії та інформацію, пов'язані з певним місцем на території КПІ ім І. Сікорського.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА РОЗРОБКИ

Метою даного проєкту є створення мобільного застосунку під операційні системи Android для супроводу по території КПІ.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- можливість визначення місцеположення за GPS;
- можливість отримати інформацію щодо поточного місця перебування у текстовому вигляді;
- наявність зручної системи налаштування додатку, яка включає в себе можливість вимкнути та ввімкнути будь-яку функцію;

					ІАЛЦ.045490.002 ТЗ	Арк.
						2
Змін	Арк.	№ докум.	Підпис	Дата		

- можливість визначити найближче місце у напрямі якого спрямований телефон;
- вивести інформацію про найближче місце напрямі якого спрямований телефон;
- вивести інформацію про місце вказане на карті.

1.2 Вимоги до апаратного забезпечення, що розробляється

- оперативна пам'ять: 1 Гб;
- наявність доступу до GPS;
- наявність доступу до мережі Інтернет (для повноцінного функціонування).

1.3 Вимоги до програмного та апаратного забезпечення користувача на етапі експлуатації

- операційна система Android;
- наявність доступу до GPS.

					ІАЛЦ.045490.002 ТЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ ПРОЄКТУВАННЯ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	04.11.2019
2.	Вивчення літератури за тематикою роботи	17.11.2019
3.	Розроблення та узгодження технічного завдання	28.11.2019
4.	Розроблення структури додатку	17.01.2020
5.	Розроблення дизайну та графічних елементів	10.02.2020
6.	Програмна реалізація додатку	15.03.2020
7.	Тестування додатку	04.04.2020
8.	Підготовка матеріалів текстової частини проєкту	24.04.2020
9.	Підготовка матеріалів графічної частини проєкту	28.04.2020
10.	Оформлення технічної документації проєкту	18.05.2020

Змін	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.002 ТЗ

Арк.

4

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			Документація загальна Новорозроблена			
A4	IАЛЦ.045490.004 ПЗ	Додаток на базі Android «Мобільний гід по університету» Пояснювальна записка	61			
A4	IАЛЦ.045490.005 Д1	Додаток на базі Android «Мобільний гід по університету» Структурна схема платформи Xamarin	1			
A4	IАЛЦ.045490.006 Д2	Додаток на базі Android «Мобільний гід по університету» Схема розроблених сервісів	1			
A4	IАЛЦ.045490.007 Д3	Додаток на базі Android «Мобільний гід по університету» Схема взаємодії компонентів інтерфейсу	1			
Zмін.	Арк.	№ докум.	Підпис	Дата	IАЛЦ.045490.003 ТП	
Rозробив	Kлимчук I. O.				Lит.	Arkysh Arkushiv
Pеревірив	Kлятченко Я.M					1 2
Kонсульта.					KPI ім. Ігоря Сікорського Кафедра CΠICKC Група KB-63	
H. контроль	Kлятченко Я.M.					
Zав. каф.	Pоманкевич B.O					

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Додаток на базі Android «Мобільний гід по університету»

Київ – 2020 року

ЗМІСТ

ЗМІСТ	1
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП.....	4
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМ ДИСТАНЦІЙНОГО МОНІТОРИНГУ	6
1.1 Особливості існуючих систем дистанційного моніторингу	6
1.2 Огляд програм-аналогів	8
1.2.1 Google Maps	8
1.2.2 Google Earth.....	16
1.3 Висновки до розділу 1	22
РОЗДІЛ 2. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ НА БАЗІ ANDROID	24
2.1 Особливості та переваги програмного інструментарію.....	24
2.1.1 Особливості мови програмування C#	24
2.1.2 Особливості мови програмування Java.....	29
2.1.3 Особливості мови програмування Kotlin.....	30
2.2 Обґрунтування вибору шаблону проєктування	31
2.2.1 MVC	31
2.2.2 MVVM	33
2.3 Опис засобів проєктування	34
2.3.1 CASE – системи.....	34
2.3.2 Мова UML	36
2.4 Висновки до розділу 2	38
РОЗДІЛ 3. ОСОБЛИВОСТІ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОБІЛЬНОГО ДОДАТКУ	39
3.1 Опис модулів ПЗ	39
3.2 Опис механізму сховища даних	41
3.3 Опис використаних компонентів	43
3.4 Особливості реалізації сервісу для розрахунку напряму.....	50

					ІАЛЦ.045490.004 ПЗ			
Змін.	Арк.	№ докум.	Підпис	Дата				
Розробив		Климчук І. О.			Додаток на базі Android «Мобільний гід по університету» Пояснювальна записка	Літ.	Аркуш	Аркушів
Перевірів		Клятченко Я.М.					1	61
						КПІ ім. Ігоря Сікорського, ФПМ, КВ-63		
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич В.О.						

3.5	Особливості реалізації сервісу розрахунку зміщення у певному напрямі	53
3.6	Особливості реалізації допоміжних сервісів	55
3.7	Рекомендації щодо подальшого вдосконалення	57
3.8	Висновки до розділу 3	58
ВИСНОВКИ.....		59
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ		60
ДОДАТКИ		

ДОДАТОК А. КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ

- ІАЛЦ.045490.005 Д1. Додаток на базі Android «Мобільний гід по університету». Структурна схема платформи Xamarin.
- ІАЛЦ.045490.006 Д2. Додаток на базі Android «Мобільний гід по університету». Схема розроблених сервісів.
- ІАЛЦ.045490.007 Д3. Додаток на базі Android «Мобільний гід по університету». Схема взаємодії компонентів інтерфейсу.
- ІАЛЦ.045490.008 Д4. Додаток на базі Android «Мобільний гід по університету». Блок схема алгоритмів розрахунку зміни координат.

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ

ДОДАТОК В. КОПІЯ ПРЕЗЕНТАЦІЇ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

- ПЗ – програмне забезпечення;
- Фреймворк (Framework) – програмна платформа, що визначає структуру програмної системи;
- Ajax – підхід до побудови користувацьких інтерфейсів веб-застосунків;
- Android – мобільна операційна система;
- API (Application Programming Interface) – прикладний програмний інтерфейс;
- GPRS – мережі з пакетної передачі даних;
- GPS – супутникова система навігації ;
- JSON – текстовий формат обміну даними;
- XAML – декларативна мова розмітки;
- MVC – схема поділу даних програми;
- MVVM – шаблон проєктування архітектури додатку;
- WPF – система для побудови клієнтських додатків Windows;

					ІАЛЦ.045490.004 ПЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Все більших обертів набуває розвиток нових технологій та їх широкого використання серед людей. У наш час ми не уявляємо свого життя без мобільного телефону. Особливо це стосується молоді, яка хоче бути обізнаною у всьому з отриманням цікавої інформації, використавши при цьому мінімум засобів. В мобільному телефоні може вміщуватися багато корисної інформації.

З кожним роком число відвідувачів у містах збільшується та збільшується оберт нових послуг. Саме з цим зростає рівень обслуговування, обумовлений потребами самих гостей міста. На сьогоднішній день використання сучасних технологій у туризмі є важливою частиною його розвитку.

Приїхавши у нове місто дуже хочеться дізнатися про всі цікаві місця, всі легенди та веселі історії. Перед поїздкою не завжди є можливість перелічувати десятки книг або шукати цікавого гіда. Декілька років назад використовували компас, паперові мапи, роздруковані каталоги та розклади. У наш час все ж таки простіше – всі речі, які нам потрібно знати для подорожі, дуже легко зберігати на смартфоні. У наш час створено багато мобільних додатків, які допомагають коротко розповісти про визначні місця й пам'ятки. А головною особливістю цих застосунків є можливість створити свій власний маршрут. Гості можуть позначати на карті місця, відстежувати події у місці і ділитися ними з друзями, робити знімки та залишати свої власні коментарі, що допоможуть іншим відвідувачам зрозуміти чи варто відвідувати це місце.

Саме тому буде корисним розроблений мною мобільний додаток на базі Android, який допоможе подорожувати по території університету з відображенням фотографій місць, які відвідує користувач. Встановивши його на мобільний телефон, планшет чи пристрій на якому стоїть ОС Android користувачі зможуть відслідкувати своє місце знаходження та переглянути

його на карті. Переглядати туристичні місця, перерахувати короткий опис про них.

Гуляючи вулицями університету цікаво, якими вони були декілька років тому.

Мета розробки – систематизування теоретичних знань отриманих впродовж навчального процесу для створення мобільного додатку з можливістю подорожувати по території університету «Київський політехнічний інститут імені Ігоря Сікорського» з відображенням фотографій, що було зроблено у цих місцях. Призначення розробки - використання проєкту у сфері мобільних додатків для подорожування по території університету фотографій, що було зроблено у цих місцях.

Завданням даної роботи є розробка додатку на базі Android для подорожування та відображення фотографій, що було зроблено у цих місцях.

					ІАЛЦ.045490.004 ПЗ	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМ ДИСТАНЦІЙНОГО МОНІТОРИНГУ

1.1 Особливості існуючих систем дистанційного моніторингу

У сучасному світі все більше набуває поширення пристроїв на базі мобільних платформ. Це обумовлено потребою людей бути постійно на зв'язку. Ще декілька років тому мобільний телефон використовували лише для зв'язку, а саме версію пересувного стаціонарного телефону, то в наш час можна робити фотографії та відео високої якості, отримувати та зберігати інформацію, відтворювати і передавати великий обсяг даних, щільно користуватися Інтернетом, виконувати функції записної книжки та грати у високоякісні ігри. Мобільний телефон є невідкладною частиною нашого життя. Без сучасного мобільного телефону не може уявити свого життя навіть дитина.

Смартфон володіє апаратними особливостями:

- більший екран, що полегшує роботу;
- швидкодіючий процесор;
- графічна карта, яка може швидко обробляти великий обсяг інформації без завантаження основного процесора;
- оперативна пам'ять, яка дозволяє керувати великими обсягами даних;
- великий обсяг пам'яті;
- Wi-Fi, GPRS.

На сьогоднішній день існує безліч мобільних платформ, основні з яких :

- Windows Phone (Windows Mobile) - це операційна система, яка містить багато можливостей, серед яких: робота з повідомленнями (SMS та електронна пошта), календар, медіа плеєр, Internet Explorer, нотатки та адресна книга. Рішення, притаманне тільки Windows Phone є відкрита архітектура, що дозволяє встановити додаткові програми і виконувати надбудови для стандартних програм. Дизайн ОС схожий на Windows, але не має з нею нічого спільного[1];

- Android - за широтою можливостей платформа Android не поступається операційним системам навіть настільних ПК. Вона являє собою багаторівневе середовище на основі ядра Linux і володіє багатими функціональними можливостями. Основу користувацького інтерфейсу складають: вікна, подання, віджети для відображення загальних елементів, таких як редаговані поля, списки і розгортаючи списки. Android володіє широким спектром можливостей підключення таким, як Wi-Fi, Bluetooth і протоколи передачі даних через мережу [2]. У стек програмного забезпечення Android входить і підтримка сервісів, заснованих на визначенні місця розташування (наприклад, GPS), і акселерометрів, проте слід зауважити, що не всі пристрої на цій платформі оснащені необхідним обладнанням;

- Symbian - на боці Symbian дружність до непідготовленого технічно користувача. Операційні системи Symbian можна розділити на дві основні підгрупи: Series 60 (як і додаткові варіанти - S80 і S90), UIQ з підтримкою сенсорного екрану і застосовується компанією SonyEricsson [3];

- iPhone OS (iOS) - iPhone OS заснована на спеціальній версії MacOS X, при цьому інтерфейс адаптований для роботи з пальцями. Особливістю апарата є великий сенсорний екран, який займає майже всю фронтальну панель і всього одна клавіша, все управління здійснюється дотиками і переміщенням пальців по екрану, все це робить смартфон як живим [4].

Ще є безліч операційних систем, хоча вони не такі популярні:

- BlackBerryOS
- PalmOS
- HPwebOS
- Bada

1.2 Огляд програм-аналогів

У наш час є безліч програм аналогів. Вони мають широкий спектр функцій. Кожна з яких має свої плюси та мінуси, але кожна в чомусь унікальна.

1.2.1 Google Maps

Google Maps - безкоштовний картографічний сервіс, розроблений компанією Google. У Google Maps є супутникові знімки, аерофотозйомку, карти вулиць, 360 ° інтерактивний панорамний вид на вулиці (перегляд вулиць), рух в режимі реального часу та планування маршрутів для подорожей пішки, машини, велосипеда та повітря або громадського транспорту. Інтерфейс Google Maps представлено на рисунку 1.1.

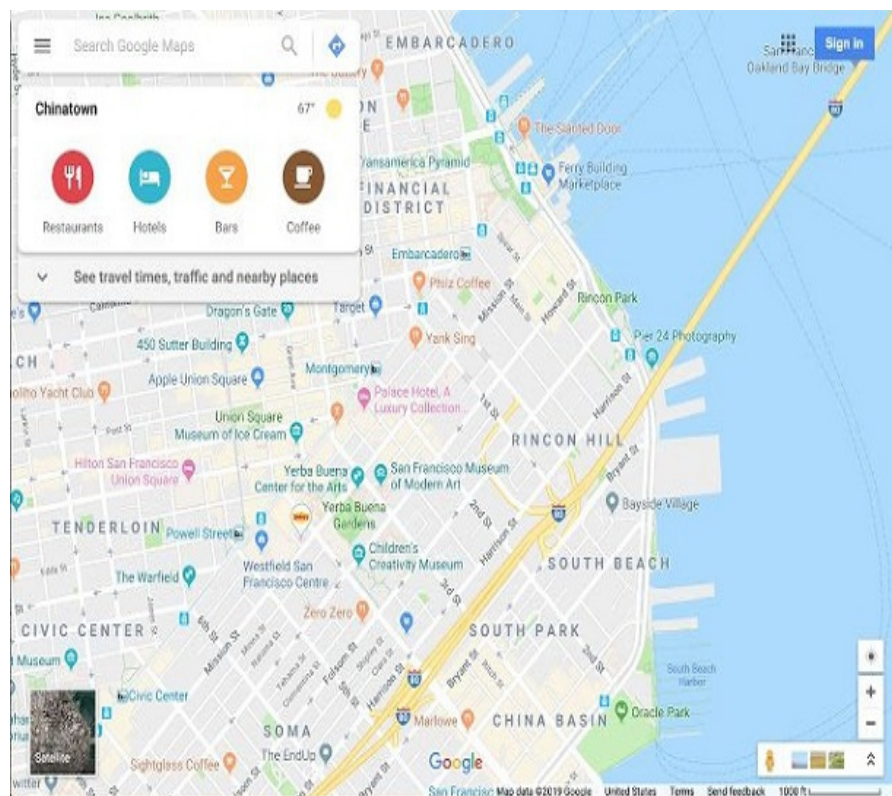


Рисунок 1.1 - Приклад вікна Google Maps

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		8

У 2020 році щомісяця Google Maps використовували понад 1 мільярд людей. Google Maps починали програму на C++ у компанії Where 2 Technologies. У жовтні 2004 року компанію придбав Google, який перетворив її на веб-додаток. Після додаткових придбань компанії з візуалізації геопросторових даних та аналізатора трафіку в режимі реального часу в Google Maps була залучена в лютому 2005 року. Попередня частина сервісу використовує JavaScript, XML та Ajax. Google Maps пропонує API, який дозволяє вбудовувати карти на сторонні веб-сайти та локатор для підприємств та інших організацій у багатьох країнах світу. Google Map Maker дозволив користувачам розширювати та оновлювати відображення сервісу у всьому світі, але з 2017 року це було припинено. Однак в Google Maps розвиток не припинився, оскільки компанія оголосила, що ці функції будуть передані програмі Google Local Guides. Супутниковий вигляд Google Maps - це "пташиний погляд". Більшість зображень міст високої якості - це аерофотозйомка, зроблена з літаків, що літають на відстані від 240 до 460 м, тоді як більшість інших зображень - із супутників. Більшість доступних знімків супутниками Google Maps оновлюються кожні три роки. Google Maps точно не могли показати області навколо полюсів, тому що використовували проекції Меркатора. Google Maps було визначено у серпні 2013 року, що це найпопулярніший додаток у світі для смартфонів. Більше половини власників смартфонів користувалися цим додатком [5].

Над Google Maps вже у 2012 році працювало понад 8200 робітників та підрядників. У 2004 році була придбана компанією Google Inc., де вона змінилася у веб-додаток Google Maps. У цьому ж місяці Google придбав компанію Keyhole, із візуалізацією геопросторових даних, чий набір додатків для погляду, Earth Viewer, у 2005 році став як найуспішніший додаток Google Earth, тоді як інші аспекти його основної технології були інтегровані в Google Maps. У вересні 2004 року Google придбав компанію ZipDash, яка здійснювала аналіз трафіку в режимі реального часу. Google Maps вперше було оголошено

про запуск в блозі Google 8 лютого 2005 року. Після урагану Катріна у вересні 2005 року, Google Maps дуже швидко оновили свої супутники для того щоб користувачі якнайшвидше побачили масштаб затоплення у різних частинах міста. Нову версію Google Maps for Mobile 2.0 було випущено 28 листопада 2007 року. Ця версія містила функцію «Моє місцезнаходження», яка використовує GPS та GPS-адресу мобільного пристрою та доповнюється визначенням найближчих мереж та сайтів зв'язку. My Location може визначити місцезнаходження користувача. 23 вересня 2008 року компанія Google оголосив про випуск додатку Google Maps для операційної системи Android. Також Google у жовтні 2009 року змінив компанію Tele Atlas як основного постачальника геопросторових даних у американській версії Maps та почав використовувати власні дані. Також потім до цієї ж версії у 2011 році додали можливість глядачеві редагувати та додавати свої зміни до Google Maps. Це дало можливість в реальному часі змінювати та отримувати оновлення не чекаючи коли сама компанія додасть оновлення. Далі компанія Google у 2012 році розпочала картографування річок та каналів Великобританії у партнерстві з Trust. Компанія пообіцяла, що почне оновлювати програму протягом року, щоб користувачі могли планувати свої поїздки, які включають у себе замки, мости та проїзді шляхи вздовж 2000 миль річковим шляхом Великобританії. Вже у 2012 році додаток Google Maps був окремо випущеним у App Store, після того, як Apple у вересні 2012 році вилучив його із програм встановленими за замовчуванням мобільної операційної системи iOS 6. Невдовзі Google Maps у 2013 році оновив свій додаток, щоб включити карту Північної Кореї. Google Maps у серпні 2013 року видалили шар, який надав посилання на вміст Вікіпедії про місцезнаходження, використовуючи геокоди Вікіпедії. Google Maps у 2014 році оновив свої дані, щоб зобразити кримську кризу. Усі інші версії Google Maps показують пунктирну межу. Також у 2015 році на карті було додано логотип Android, що мочиться на логотип Apple. Через короткий період ця проблема була усуненою і через це компанія Google публічно вибачився. Через

					ІАЛЦ.045490.004 ПЗ	Арк.
						10
Змін.	Арк.	№ докум.	Підпис	Дата		

це Google відключив можливість користувачам вносити зміни по всьому світу. Поки не зможуть розробити політику, щоб уникати вандалізму. Компанія Google у 2015 році випустила нові знімки із супутника по всьому світі отримані від з Landsat 8, що містив понад 700 трильйонів пікселів нових даних. Не довго чекаючи Google Maps у 2016 році купили аналітичні карти для запуску Urban Engines. Південна Корея у 2016 році запропонувала компанії Google зробила умовний доступ, який дозволяє корейським постачальникам карт розробляти детальні карти. Але Google відмовився від такої пропозиції, бо вважали, що це знизить якість. Також оновлення додало прямий доступ до зображень Місяця та Марса. У 2017 році серед популярних додатків, включаючи YouTube, Chrome, Gmail та Google Play, Google Maps мав 1 мільярд користувачів щомісяця і став найпопулярнішим додатком. Google Maps не зупинявся у розвитку свого додатку і ось вже у серпні 2018 року було оновлено для показу тривимірної кулі. Але також можна повернутися до двовимірної. Для мобільних пристроїв цей додаток випустили тільки у вересні 2018 року та містили GPS навігацію із спеціальними можливостями допомоги при парковці. У 2018 році вирішили ввести зміни у структурі API, це допомогло об'єднати 18 різних кінцевих точок у три сервіси [6].

Google Maps програма на мові C++, яка була розроблена двома програмістами братами Ларсом та Єнсом Ейлstrupом Рфсмуссенем у компанії «Where 2 Technologies». Спочатку цей додаток був розроблений для того, щоб завантажували його окремо, але пізніше вирішили, щоб передати цей веб-додаток компанії Google. У січні 2019 року Google Maps додали швидкі пластинки та сповіщення камери. Google Maps – це планувальник маршрутів, який дозволяє своїм користувачам знаходити доступні вказівки під час водіння, громадського транспорту, прогулянок чи їзди на велосипеді. Google співпрацює у всьому світі з понад 800 провайдерами громадського транспорту, для того щоб прийняти GTFS. У додатку також є зірочки для оцінювання якихось закладів, це допомагає іншим користувачам. У останньому оновленні додатку є

спеціальний режим – навігація без очей. Також компанія Google впровадила нові оновлення, щоб відображати швидкість руху, за допомогою GPS-локацій користувачів мобільних телефонів у реальному часі. Для безпеки користувачів Google зробив, щоб вся інформація про швидкість та місцезнаходження залишається анонімною. Ці налаштування доступні кожному телефону, але за бажанням є функція поділитися інформацією про своє місцезнаходження. Вже у 2017 році компанія Google випустила Google Street View – нова функція, яка надає 360° панорамних видів вулиць різних локацій. Але не всім містам було доступним, лише п'ять міст США. З того часу ця функція розширилася по всьому світу. Функція Street View викликала багато суперечок після його виходу через проблеми конфіденційності нецензурного характеру панорамних фотографій. Фотографії робляться у громадських місцях, тому обличчя людей та номерні знаки почали розмивати. Вже у 2014 році Google запускає проєкт Google Underwater Street View, який включав 2300 кілометрів Великого Бар'єрного рифу в Австралії в 3D форматі. Ці знімки були зробленими спеціальними камерами, які можуть обертатися на 360 градусів і роблять знімки кожні 3 секунди. У 2017 році на Google Maps і в Google Earth стала доступна функція новітньої з вулиці у внутрішніх просторах Міжнародної космічної станції. У грудні 2009 року компанія Google представляє новий вигляд, що складалася із знімків під кутом 45°, наче «пташиний погляд». Вперше це було доступно у містах Сан-Хосе та Сан-Дієго. Але спочатку ця функція була доступна лише розробникам через API Google Maps. Ця функція у лютому 2010 році була представлена як експериментальна функція у Google Maps Labs. Зображення на 45° стають доступними на Google Maps у деяких містах США та у всьому світі. Google співпрацює списки підприємств із кількох он-лайн та офлайн-джерел. Для зменшення дублювання в індексі алгоритм Google автоматично поєднує списки автоматично на основі адреси, номера телефону або геокоду, але іноді інформація для окремих підприємств буде ненавмисно об'єднана між собою, що призводить до того, що у списках неточно

					ІАЛЦ.045490.004 ПЗ	Арк.
						12
Змін.	Арк.	№ докум.	Підпис	Дата		

розміщенні елементи з кількох підприємств. Google дозволяє власникам бізнесу перевірити свої власні дані за допомогою Google My Business, а також набрати волонтерів для перевірки чи виправленні основних даних. Google Maps легко можуть управляти підприємствами, які не знаходяться у тій місцевості. Але іноді цим зловживають, щоб наздогнати свою конкуренцію, розміщуючи не перевірені списки на сайтах, що інформація перенесеться на Google (копії сайтів). Люди, які оновлюють ці списки, не користуються своїм ім'ям та ім'ям компанії. Ключові слова та місцезнаходження розміщують у Google Maps, Наприклад, в Австралії компанії почали помічати популярність неправильних списків підприємств у різних галузях. Скоро Google Maps у 2011 році додали картки закритих приміщень, що дозволяє орієнтуватися у будинках, а саме: аеропорти, музеї, торгові центри, великі магазини, університети, транзитні станції та інші громадські приміщення. Google запропонував власникам громадських приміщень надати плани своїх поверхонь. Також суди входить різні поверхні будівлі чи станції метро, натискаючи на селектор рівня, який відображається біля будь-яких структур, на яких зображено декілька рівнів [7].

Мої карти - це функція у Google Maps, випущена у 2007 році, яка дозволила користувачам створювати свої власні карти для свого особистого користування. Користувачі також можуть додати точки до Google Maps за допомогою редактора WYSIWYG. Цей додаток також доступний для Android з березня 2013 року під назвою Google Maps Engine Lite. Google Local Guides - це програма, випущена Google Maps, щоб користувачі мали можливість робити свої внески у Google Maps, і це додаткова для них перевага. Програма частково є наслідником Google Map Maker, так як останні програми стали змінюватися у додаток. Додаток Maps додав можливість змінювати тему свого режиму. Коли користувач перетягує карту, квадрати сітки завантажуються з сервера та вставляються на сторінку. Коли користувач здійснює пошук компанії, результати завантажуються у фоновому режимі для вставки на бічну панель та карту. Саме з поданням форми iframe (прихований кадр) використовується, так

					ІАЛЦ.045490.004 ПЗ	Арк.
						13
Змін.	Арк.	№ докум.	Підпис	Дата		

як він зберігає історію браузерів. Як і інші додатки Google, Google Maps використовують JavaScript. Сайт використовує JSON, а не XML для передачі за можливістю більшої продуктивності. У жовтні 2011 року компанія Google оголосила MapsGL, як версією Maps з кращою візуалізацією та кращими переходами. Google Maps використовують JPG, PNG, PDF, BMP чи GIF. Користувачі облікового запису Google, можуть зберігати свої місця, накладаючи на карту різні кольорові позначки. Ці "Збережені місця" можна додавати та впорядковувати в списку з різними іменами. Також можна ділитися цими позначками з іншими користувачами. Компанія Google купила дані локальних карт у створених компаній або укласти договір оренд, але дані захищені авторським правом. Карти вулиць в Японії здаються в оренду від Zenrin. Карти вулиць в Китаї здаються в оренду від AutoNavi. Російські карти вулиць здаються в оренду у «Геоцентр Консалтинг» і «Теле Атлас». Дані для Північної Кореї отримують із супровідного проєкту Google Map Maker. Але карти у деяких районах можуть не відповідати супутниковим знімкам. Дані на вулиці можуть бути абсолютно помилковими або просто застарілими: "Найбільша проблема - це валюта даних, достовірність даних", - сказав представник Google Earth Брайан Макклendon. Це призвело до додання функція в березні 2008 року редагування місць розташування усіх будинків та підприємств. Але на Google Maps було встановлено обмеження через цензуру місць, які вважали потенційними загрозами. В деяких випадках область редагування призначена для конкретних будівель, але в інших випадках, таких як Вашингтон, округ Колумбія, обмеженням є використання застарілих зображень.

У 2015 році Google запустив API Google Maps, щоб дозволити розробникам перетворювати Google Maps у свої веб-сайти. Ця послуга є безкоштовною і не потрібен ключ API, але це було до червня 2018 року. Після ж для доступу API потрібно мати ключ API, пов'язаний із обліковим записом Google Cloud. Більше ніж 1 000 000 веб-сайтів використовують API Google

					ІАЛЦ.045490.004 ПЗ	Арк.
						14
Змін.	Арк.	№ докум.	Підпис	Дата		

Maps, але вже у вересні 2011 року Google оголосив, що скасовує API Google Maps для Flash. API Google Maps є безкоштовним для комерційного використання, але за умови, що сайт, на якому він використовується є доступним усім і безкоштовним. Але сайти не змогли відповідати цим вимогам, тому можуть придбати API Google Maps лише для бізнесу.

З 2018 року Google вирішив підняти ціни API та почав вимагати платіжного профілю. Хоча у Китаї обмеження географічних даних Google Maps повинні співпрацювати з китайськими постачальниками. З 2006 року цим партнером є AutoNavi. У Китаї Державна рада залучає всі карти використовувати у системній координаті GCJ-02, яка зовсім зрушена від системи WGS-84, що використовується майже у всіх країнах світу. Також WGS-84 для супутникових знімків, викликало проблему зсуву GPS в Китаї. Спроба вирівнювання меж містить відмінності між google.cn/Maps та google.com/Maps. На картах межі позначаються пунктиром, що вказує на суперечки районів чи кордонів. Але google.cn показує кордон суворо відповідно до китайських стверджень, без пунктирів. Регіон Південного Тибету, на який претендує Китай, але Індія є під управлінням як значна частина Аруначал-Прадеш, відображається на китайському кордоні через google.cn,. Також Google.cn зображає Тайвань та Південно-Китайські острови як частину самого Китаю. На картах Google Ditu уже не омиває великих державних органів, таких як Президентський палац, п'ять юанів та Верховний суд. Google.cn/Maps не містить «Мої карти». З іншого боку, google.cn відображає практично весь текст китайською мовою, а google.com/Maps відображає більшість текстів (реальний текст, обраний користувачем, а також текст на карті) англійською мовою. Така поведінка відображення англійського тексту не є послідовною, але переривчастою - іноді це англійською, інколи - китайською. Критерії вибору мови відображення публічно не відомі.

1.2.2 Google Earth

Google Earth - це комп'ютерна програма, яка надає 3D-зображення Землі знімків, які зроблені із супутника. Програма відображає Землю, накладаючи на 3D-глобус супутникових знімків, що дозволяє користувачам бачити міста з різних куточків країни. Приклад 3D-зображення Землі представлено на рисунку 1.2.



Рисунок 1.2 - Приклад 3D-зображення Землі

Користувачі можуть вивчати нашу Землю, вводячи адреси чи координати, або використовуючи клавіатуру чи мишу. Програму також можна завантажити на смартфон або планшет, використовуючи сенсорний екран або стилус для навігації. Користувачі можуть використовувати програму для додавання власних даних за допомогою мови розмітки Keyhole та завантаження їх через різні джерела. Google Earth здатна показувати різні види зображень, накладених на земну поверхню, а також є клієнтом служби веб-карт. Вони створили Keyhole, Inc. на чолі з Джоном Хенке. Потім Кейхоле розробив спосіб

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

16

передавати великі бази даних відображення даних через Інтернет клієнту програмне забезпечення, ключова частина технології, та придбані печворки відображення даних урядів та інших джерел. Компанія Google виявив, що Google Earth охоплює понад 98 відсотків у світі, і вона захопила 10 мільйонів миль зображень Street View, відстань, яка може об'їхати земну кулю більше 400 разів. Крім навігації по Землі, Google Earth пропонує ряд інших інструментів через додаток. Доступні додаткові глобуси для Місяця та Марса, а також інструмент для перегляду нічного неба. Гра в симулятор польоту також включена. Інші функції дозволяють користувачам переглядати фотографії з різних місць, завантажених у Panoramio, інформацію, яку надає Wikipedia в деяких місцях, і зображення перегляду вулиць. Веб-версія Google Earth також включає Voyager - функцію, яка періодично додає тури за програмою, які часто представляються вченими та документалістами. Деякі розглядали Google Earth як загрозу конфіденційності та національній безпеці, що призводить до того, що програма буде заборонено в багатьох країнах. Деякі країни попросили зафіксувати певні райони в супутникових знімках Google. Основна технологія Google Earth спочатку була розроблена в компанії Intrinsic Graphics наприкінці 1990-х. У той час компанія розробляла бібліотеки програмного забезпечення для 3D-ігор. Як демонстрація їх 3D-програмного забезпечення, вони створили глобус, на який можна збільшувати масштаб. Продукт під назвою "Keyhole EarthViewer" продавався на компакт-дисках для використання в таких сферах, як нерухомість, містобудування, оборона та розвідка. Користувачі сплачували щорічну плату за послугу. Незважаючи на укладення різних угод з Nvidia та Sony, невелика компанія з усіх сил намагалася заробити зарплату, а співробітники виїжджали. Компанії почало щастити на початку 2003 року, коли CNN отримала знижку на програмне забезпечення в обмін на розміщення логотипу Keyhole в ефірі, коли карта була використана. Кейчол не сподівався, що це буде більш ніж короткі 5- або 10-секундні попередньо записані анімаційні кліпи, але він широко використовувався Майлом О'Браєном у прямому ефірі під

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

17

час вторгнення в Ірак 2003 року, що дозволило CNN та мільйонам глядачів стежити за ходом війни таким чином, якого раніше ніколи не бачили. Суспільний інтерес до програмного забезпечення вибухнув. Незабаром з Keyhole звернулися до фірми Центрального розвідувального агентства з венчурним капіталом In-Q-Tel та Національного агентства геопросторової розвідки для використання у базах даних оборонного картографування, що дало Keyhole так потрібний грошовий вплив. Intrinsic Graphics був проданий у 2003 році до Vicarious Visions після того, як його ігрові бібліотеки не продавались добре, а його основна група інженерів та менеджменту перейшла до Keyhole, а Ханке залишився на чолі. У той час Google виявляв, що понад 25% його пошуків носили геопросторовий характер, включаючи пошук карт та напрямків. У жовтні 2004 року компанія Google придбала Keyhole як частину стратегії для кращого обслуговування своїх користувачів. Зображення Google Earth відображаються на цифровому глобусі, який відображає поверхню планети за допомогою єдиного складеного зображення з далекої відстані. Після збільшення масштабу зображення переходить у різні зображення тієї області до іншої. Зображення отримують із супутників. До запуску НАСА та супутника Landsat 8 USGS US частково покладався на зображення Landsat 7, які постраждали від апаратної несправності.

У 2013 році Google використовував обмін даними для виправлення проблеми, надаючи те, що було описано як спадкоємця зображення "Синього мармуру Землі", з єдиним великим зображенням всієї планети. Це було досягнуто шляхом комбінування наборів зображень, знятих із Landsat 7, щоб усунути хмари та діагональні прогалини, створивши єдине «мозаїчне» зображення. Зараз Google використовує Landsat 8 для надання зображень у більш високій якості та з більшою частотою. Зображення розміщується на серверах Google, на які додаток зв'язується програмою при відкритті та вимагає підключення до Інтернету. Роздільна здатність зображень коливається від 15 метрів до 15 сантиметрів. Більшу частину Землі Google Earth використовує дані

цифрової моделі висоти, зібрані місією NASA Shuttle Radar Topography. Це створює враження тривимірного рельєфу місцевості, навіть якщо зображення лише двовимірне. Google запевняє, що кожне зображення, створене з Google Earth за допомогою супутникових даних, наданих Google Earth, є захищеною авторським правом картою. Будь-яка похідна інформація від Google Earth створена з даних, на які Google заявляє про авторські права відповідно до закону про авторські права США. Google надає ліцензії на ці дані, дозволяючи, між іншим, некомерційним особистим використанням зображень (наприклад, на особистому веб-сайті чи в блозі), якщо зберігаються авторські права та атрибути. На відміну від цього, у зображеннях, створених за допомогою програмного забезпечення WorldWind НАСА, використовуються зображення Blue Blue Marble, Landsat або USGS, кожне з яких знаходиться у відкритому доступі. У версії 5.0 компанія Google захотіла представити «Історичне зображення», що дозволяє користувачам переглядати зображення, які були зроблені давно. Ця функція дозволяє спостерігати за змінами району в часі. Використання функції тимчасового відключення дає можливість переглядати масштабоване відео аж до 32 років. Google Earth показує тривимірні моделі будівель у деяких містах, включаючи фотореалістичні 3D-зображення. У червні 2012 року Google оголосив, що замінить створені користувачем 3D-будівлі автоматично створеною 3D-сіткою. Це буде поетапно, починаючи з окремих міст, за винятком таких міст, як Лондон та Торонто, яким потрібно було більше часу для обробки детальних зображень їх величезної кількості будівель. Причина - більша рівномірність у 3D-будівлях та конкурувати з Nokia Here та Apple Maps, які вже використовували цю технологію. Покриття розпочалося в тому році в 21 місті чотирьох країн. На початку 2016 року 3D-зображення було розширено до сотень міст понад 40 країн, включаючи кожен штат США і охоплювало кожен континент, крім Антарктиди. У 2009 році у співпраці між Google та Museo del Prado в Мадриді музей відібрав 14 своїх картин для фотографування та відображення з роздільною здатністю 14 000 мегапікселів

					ІАЛЦ.045490.004 ПЗ	Арк.
						19
Змін.	Арк.	№ докум.	Підпис	Дата		

всередині 3D версії Prado в Google Earth та Google Maps. 15 квітня 2008 року з версією 4.3 Google повністю інтегрував перегляд вулиць у Google Earth. Street View відображає 360° панорамні фотографії на рівні вулиць окремих міст та їх околиць. Фотографії були зроблені камерами, встановленими на автомобілях, їх можна переглядати в різних масштабах і з багатьох кутів, а також орієнтуватися на накладені на них піктограми стрілок. Використовуючи Street View на Google Earth, користувачі мають можливість відвідувати та досліджувати 30 об'єктів світової спадщини ЮНЕСКО з історичним контекстом та шпильками для кожного з них.

Введена в Google Earth 5.0 у 2009 році, функція Google Ocean дозволяє користувачам збільшувати масштаб нижче поверхні океану та переглядати 3D батиметрію. Підтримуючи понад 20 шарів змісту, він містить інформацію провідних вчених та океанографів. Також 14 квітня 2009 року Google додав батиметричні дані для Великих озер. У червні 2011 року Google збільшив роздільну здатність деяких районів глибокого океану з 1 кілометрових сіток до 100 метрів. Характеристики високої роздільної здатності були розроблені океанографами в обсерваторії Землі Ламон-Догерті Колумбійського університету з наукових даних, зібраних на дослідницьких круїзах. Більш чіткий фокус доступний приблизно для 5 відсотків Світового океану. Це можна побачити на Гудзоні біля Нью-Йорка, на узбережжі Віні поблизу Гаваїв та хребті Мендочіно біля узбережжя Тихого океану США. Google Earth має численні функції, які дозволяють користувачеві дізнатися про конкретні місця. Вони називаються "шарами" і включають різні форми засобів масової інформації, включаючи фото та відео. Деякі шари містять тури, які орієнтують користувача між певними місцями у встановленому порядку. Шари створюються за допомогою мови розмітки Keyhole або KML, яку користувачі також можуть використовувати для створення спеціальних шарів. Місця розташування можуть бути позначені позначками місця та впорядковані в папках. Наприклад, користувач може використовувати орієнтири, щоб

перелічити цікаві орієнтири по всьому світу, а потім надати опис із фотографіями та відео, які можна переглянути, натиснувши на позначки місць під час перегляду нового шару програми. У грудні 2006 року Google Earth додала нову інтеграцію з Wikipedia та Panoramio. Також є шар спільноти з проєкту Wikipedia-World. Використовується більше координат, на дисплеї є різні типи, підтримуються різні мови, ніж вбудований шар Wikipedia. У шарі Panoramio розміщені зображення, завантажені користувачами Panoramio, розміщені в Google Earth на основі даних про місцезнаходження, наданих користувачем. Окрім плоских зображень, Google Earth містить також шар для представлених користувачем панорамних фотографій, які можна переміщуватись аналогічно перегляду вулиць. Google Earth містить кілька функцій, які дозволяють користувачеві стежити за поточними подіями. У 2007 році Google почав пропонувати користувачам можливість відстежувати дані про трафік, які надає Google Traffic в режимі реального часу, на основі інформації, що переповнюється в GPS-адресах користувачів мобільних телефонів. У програмі Google Earth 4.2 до програми додано льотний симулятор. Спочатку це було прихованою функцією, коли вона була представлена в 2007 році, але, починаючи з 4.3, їй було надано опцію в меню. Крім керування клавіатурою, тренажер можна керувати за допомогою миші або джойстика. Симулятор також працює з анімацією, що дозволяє об'єктам, таким як літаки, анімувати під час роботи на тренажері. Інший симулятор польоту, GeoFS, був створений під назвою GEFS-Online за допомогою додатка API Google Earth для роботи в веб-браузері. З 1 вересня 2015 року програма тепер використовує програму CesiumJS з відкритим кодом, через припинення роботи плагіну Google Earth. Рідка галактика - це сукупність комп'ютерів під керуванням Google Earth, що створює вражаючий досвід. 30 вересня 2010 року Google оприлюднив конфігурацію та схеми для своїх установок, розмістивши код та посібники з налаштування на вікі «Рідка галактика». Рідка галактика також використовувалася як панорамний

					ІАЛЦ.045490.004 ПЗ	Арк.
						21
Змін.	Арк.	№ докум.	Підпис	Дата		

переглядач фотографій за допомогою KRpano, а також переглядач Google Street View за допомогою Peruse-a-Rue - метод синхронізації декількох клієнтів API Maps. Google Earth випущена на комп'ютерах macOS, Linux, iOS та Android. Версія Linux почалася з бета-версії Google Earth 4, як рідного порту за допомогою інструментарію Qt. Фонд вільного програмного забезпечення розглядає розробку безкоштовного сумісного клієнта для Google Earth як пріоритетний проєкт вільного програмного забезпечення. Google Earth була випущена для Android 22 лютого 2010 року та iOS 27 жовтня 2008 року. Мобільні версії Google Earth можуть використовувати жести для переміщення по земній кулі, збільшення або поворот перегляду та дозволяють вибрати поточне місцеположення. Автомобільна версія Google Earth стала доступною в Audi A8 2010 року. 27 лютого 2020 року Google відкрив свою веб-версію Землі для таких браузерів, як Firefox, Edge та Opera [8].

1.3 Висновки до розділу 1

У даному розділі було проаналізовано сучасні операційні системи, які використовуються для мобільних пристроїв. Було визначено, що найбільш розповсюдженою платформою для смартфонів є Android, оскільки такі операційні системи, як iOS, можуть працювати лише на певних пристроях (iPhone), у той час, як платформа Android може працювати майже на будь-якому смартфоні, оскільки побудована на основі операційної системи Linux.

Також було проаналізовано основні аналоги. Було визначено, що немає програм, які можуть бути використані в якості гідів за територією університету. Сучасні аналоги карт (Google Maps, iPhone Maps) не надають інформацію про історію того чи іншого місця і спрямовані більше на визначення шляху до певної локації. Такі додатки, як Google Earth, дозволяють бачити зображення певного району, які були зроблені давно, але теж мають недоліки, такі, як відсутність покриття багатьох територій світу та відсутність історичної довідки.

					ІАЛЦ.045490.004 ПЗ	Арк.
						22
Змін.	Арк.	№ докум.	Підпис	Дата		

Отже, для реалізації було обрано платформу Android, яка є більш розповсюдженою. Також було визначено основні недоліки наявних аналогів і сформовано вимоги до нової системи мобільного гідів за територією університету.

					<i>ІАЛЦ.045490.004 ПЗ</i>	Арк.
						23
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ НА БАЗІ ANDROID

2.1 Особливості та переваги програмного інструментарію

2.1.1 Особливості мови програмування C#

C# – об'єктно-орієнтована мова програмування високого рівня, яка, окрім об'єктно-орієнтованого підходу, підтримує більшість сучасних парадигм, таких, як функціональне програмування, узагальнене програмування, подійно-орієнтоване програмування тощо.

C# є C-подібною мовою програмування, найбільшим схожим синтаксисом володіють Java та C++. Мова має строгу статичну типізацію, разом з тим має можливості для динамічної типізації (починаючи з версії C# 5.0).

Мова була розроблена під егідою компанії Microsoft у 1999-2001 роках і починаючи зі своєї першої версії була однією з основних мов для розробки застосунків на платформі .NET. Для проєктування нової мови було залучена велика кількість спеціалістів, зокрема Андерса Гейлсберга (який створив такі мови, як Turbo Pascal та Delphi). Основна мета була в тому, щоб спираючись на досвід попередніх мов, таких, як Java, Delphi, C++, взяти з них сильні сторони, та позбавитись від концепцій, які зарекомендували себе як проблемні. Як приклад таких концепцій можна взяти множинне спадкування з C++, яке частково було замінене інтерфейсами, або примітивні типи у Java, які було замінені ValueType у C#, які є спадкоємцями базового типу Object.

Сьогодні мова досить активно розвивається, як і платформа .NET. У 2015 році була випущена нова реалізація платформи .NET — .NET Core версії 1.0. Основною перевагою нової реалізації є кросплатформенність, пакетний менеджер Nuget. Разом з тим, код компілюється за допомогою компілятора RyuJIT, як і у платформі Java, у проміжний байт код, який виконується на віртуальній машині CoreCLR. Це дозволяє переносити програму на будь-яку

машину, яка має CoreCLR, без перекомпіляції а також писати модулі для одного застосунку різними мовами, які відповідають стандарту .NET [9].

Остання версія .Net Core 3.1 (має статус LTS – Long time serviced) вийшла у грудні 2019 року. Ця версія має підтримку таких технологій, як WPF, UWP, Windows Forms, ASP.NET Core. Велика кількість проєктів, яка була реалізована на платформі .NET Framework мігрує на нову версію .NET Core.

Серед важливих можливостей, які було реалізовано корпорацією Microsoft у мові програмування C#, варто зазначити такі:

- компонентно-орієнтований підхід до програмування (який характерний і для ідеології Microsoft. NET в цілому);
- строга типізація, яка поширюється і на динамічну типізацію за допомогою оператора var;
- модель інтерфейсів, яка дозволяє одному класу реалізовувати багато різних наборів поведінок;
- мова запитів LINQ, яка дозволяє за допомогою простих і коротких команд організовувати роботу з наборами даних різної складності;
- обробка помилок за допомогою блоків коду try/catch/finally;
- підтримка збірки сміття, яка дозволяє програмісту не піклуватись про керування пам'яттю, що зменшує кількість помилок;
- підтримка доступу до некерованої пам'яті;
- шаблони (generics), які дозволяють підтримувати однакову поведінку для різних типів, разом з тим зберігаючи безпеку типів та обмеження на типізацію;
- асинхронні операції за допомогою ключових слів async/await та типу Task;
- багатопотоковість за допомогою бібліотеки TPL, та колекції, які підтримують безпечні для потоків операції;
- мультипарадигменність мови.

.NET – A unified platform



Рисунок 2.1 - Структура платформи .NET

Для створення мобільних додатків на платформі .NET існує технологія Xamarin. Вона є розвитком старішої технології, розробкою та розвитком якої займалась компанія Mono, яка була куплена компанією Microsoft у 2016 році. Вихідний код Xamarin SDK відкритий, що забезпечує швидкий розвиток платформи та широку спільноту розробників, що нею користуються.

Одним з фреймворків платформи Xamarin є Xamarin.Forms. Це UI фреймворк, який дозволяє будувати інтерфейси з застосуванням однієї кодової бази для платформи Android, iOS та Universal Windows Platform (для пристроїв на основі платформи Windows).

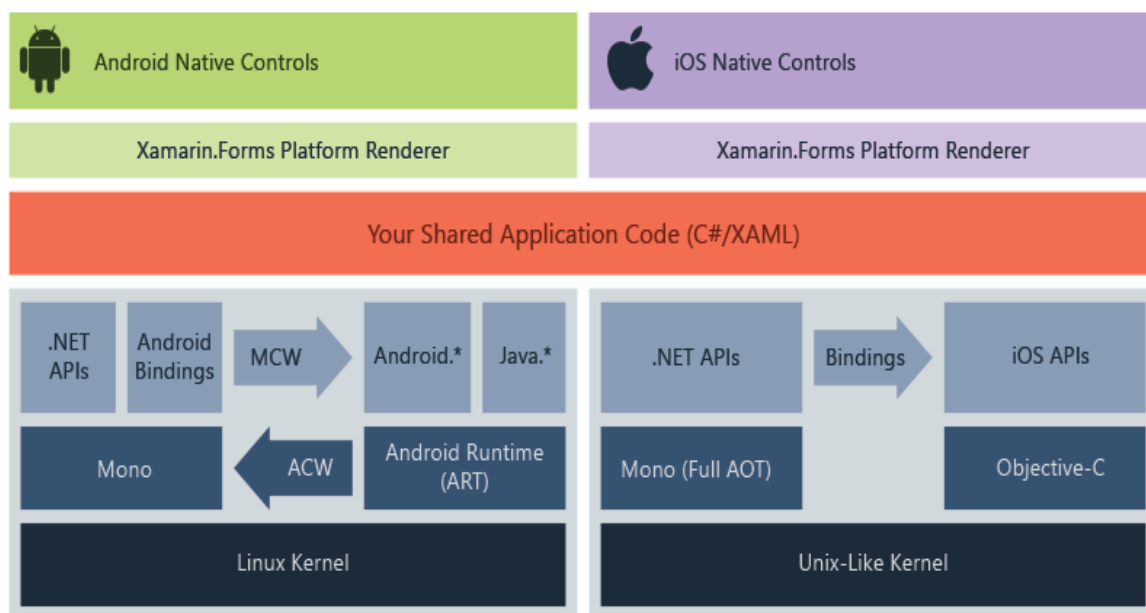


Рисунок 2.2 - Структура роботи платформи Xamarin.Forms

Переваги використання Xamarin.Forms:

- в процесі розробки створюється єдиний код для всіх платформ;
- Xamarin надає прямий доступ до нативних API кожної платформи;
- при створенні додатків ми можемо використовувати платформу .NET і мову програмування C# (а також F#), яка є досить продуктивною, а також досить простою у застосуванні;
- Xamarin.Forms підтримує кілька платформ. Основні платформи: Android, iOS, UWP, Tizen. Додаткові платформи в стані прев'ю: MacOS, WPF, GTK#.

Xamarin.Forms надає єдиний API для створення елементів інтерфейсу на платформах. Цей API може бути реалізований у XAML або C# і підтримує прив'язку даних для таких моделей, як Model-View-ViewModel (MVVM).

Під час виконання Xamarin.Forms використовує візуалізатори платформи для перетворення елементів кросплатформенного інтерфейсу в нативні елементи керування на Android, iOS та UWP. Це дозволяє розробникам бачити нативний інтерфейс і одночасно отримувати переваги використання коду для кросплатформенної розробки.

Програми Xamarin.Forms зазвичай складаються із спільної бібліотеки .NET Standard та окремих проєктів платформи. Бібліотека, що поділяється, містить представлення XAML або C# та будь-яку бізнес логіку, моделі, сервіси. Проєкти платформ містять будь-яку логіку платформи або пакети, необхідні додатку.

Xamarin.Forms використовує Xamarin для запуску програм .NET на власних платформах.

Разом з тим, головним недоліком платформи Xamarin є певна обмеженість її використання для розробки додатків з використанням одного коду. Це зумовлено тим, що файлова система, життєвий цикл додатків та структура різних операційних систем сильно відрізняються, і для деяких сутностей неможливо побудувати однаковий інтерфейс.

Саме для таких випадків використовується бібліотека Xamarin.Essentials. Xamarin.Essentials - це бібліотека, яка надає кросплатформенні API для функцій нативного пристрою. Як і сам Xamarin, Xamarin.Essentials - це абстракція, яка спрощує процес доступу до нативних утиліт. По суті, бібліотека є адаптером для нативних функцій певної операційної системи і дозволяє викликати ці функції з коду на C#. Деякі приклади утиліт, що надаються Xamarin.Essentials, включають:

- інформацію про пристрій;
- файлову систему;
- акселерометр;
- набір номера телефону;
- переклад тексту;
- блокування екрану.

2.1.2 Особливості мови програмування Java

Java – об’єктно-орієнтована мова програмування високого рівня зі строгою типізацією, яка була розроблена компанією Sun Microsystems (була придбана компанією Oracle). Мова входить до родини С-подібних мов. Зараз розробка ведеться спільнотою, організованою через Java Community Process, компоненти, написані цією мовою, розповсюджуються під ліцензією GPL.

Перевагами Java є:

- автоматичне керування пам'яттю (використання збірника сміття);
- розширені можливості обробки виняткових ситуацій;
- багатий набір засобів фільтрації введення-виведення;
- набір стандартних колекцій: масив, список, стек і т. п.;
- наявність простих засобів створення мережових додатків (у тому числі з використанням протоколу RMI);
- наявність класів, що дозволяють виконувати HTTP-запити і обробляти відповіді;
- вбудовані в мову засоби створення багатопотокових додатків, які потім були перенести на багато мов (наприклад Python);
- уніфікований доступ до баз даних;
- на рівні окремих SQL-запитів - на основі JDBC, SQLJ;
- на рівні концепції об'єктів, що володіють здатністю до зберігання в базі даних - на основі Java Data Objects і Java Persistence API;
- підтримка узагальнень (починаючи з версії 1.5);
- підтримка лямбда, замикань, вбудовані можливості функціонального програмування (з 1.8);
- наявність великої кількості написаних бібліотек та великої кількості інформації про мову в мережі.

Серед недоліків слід зазначити:

- досить повільний розвиток;

- наявність скалярних типів даних, для яких довелося вводити штучні «обгортки»;
- відсутність стандартних шаблонів для проєктів;
- відсутність автоматичного пакетного менеджера, необхідно налаштовувати пакети через xml;
- визначення констант через ключове слово `final`, що вносить певну плутанину;
- всі методи за замовчуванням є віртуальними;
- відсутність структур;
- відсутність багатовимірних масивів (існують тільки зубчаті масиви);
- відсутність одного батьківського класу для всіх типів виключних ситуацій;
- відсутність беззнакових цілих типів;
- недоліки розрахунків операцій з плаваючою комою.

Як можна помітити, багато в чому мова C# зобов'язана мові Java, оскільки вони мають багато спільних рис. У той же час, багато недоліків, які були присутні у Java прибрані у C# [10].

Необхідно зазначити, що мова Java є стандартом для написання застосунків на платформі Android, оскільки заснована на ядрі операційної системі Linux, і реалізації JVM. Тобто, програми, написані на Java є нативними для ОС Android.

2.1.3 Особливості мови програмування Kotlin

Kotlin – об'єктно-орієнтована мова програмування зі статичною типізацією, яка виконується поверх JVM. Основною задачею розробників мови було створити мову, яка була б лаконічніша за Java та простіша за мову Scala. Основною перевагою Kotlin перед Scala стала її повна сумісність з мовою Java [11].

Основні переваги мови Kotlin:

- компілюється в байткод JVM або в JavaScript;
- програми можуть використовувати всі існуючі Java-фреймворки і бібліотеки. Kotlin можна інтегрувати з Maven, Gradle і іншими системами збірки;
- мова дуже проста для вивчення;
- вихідний код відкритий;
- в IntelliJ доступна автоматична конвертація Java-коду в Kotlin і навпаки;
- мова null-безпечний - відсутність NullPointerException, замість цього помилки цього типу відловлюються на етапі компіляції;
- синтаксис, який легко читається.

Основні недоліки мови:

- мова досить молода, менша спільнота розробників, у порівнянні з мовою Java;
- суттєва відмінність синтаксису від Java;
- відсутність стандартизації.

Мова Kotlin стала стандартом для розробки на платформі Android у 2019 році.

2.2 Обґрунтування вибору шаблону проектування

2.2.1 MVC

Model-View-Controller (MVC) – архітектурний шаблон, який знайшов широке використання в процесі проектування та розробки програмного забезпечення. Шаблон поділяє систему на 3 рівні: модель (об'єкти системи), відображення та керування. Кожна з цих структур може змінюватись незалежно одна від одної, при цьому зміни в інтерфейсі користувача не впливають на керування даними, зміни в моделі не впливають на відображення даних [12].

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

31

Основна мета шаблону – забезпечення гнучкого дизайну програмної системи, при якому кожна частина може розроблятися різними людьми. Також це спрощує подальші зміни та розширення програм і надає можливість повторного використання коду. Якщо система досить велика, даний шаблон дозволяє зробити її більш структурованою за допомогою цього поділу.

Розглянемо детальніше кожен з рівнів.

Модель – представляє собою дані у системі. Зазвичай є простими об'єктами, які не вміщують в собі логіку, хіба що логіку валідації полів. Зазвичай вирізняють різні рівні моделей: доменні моделі та моделі передачі даних. Доменні моделі відображають структуру бази даних, або файлу, де дані зберігаються. Моделі передачі даних – об'єкти, які бачить користувач.

Необхідність такого розділення корисна за наступних причин:

- деякі елементи моделей, наприклад, ідентифікатори, користувач не має бачити;
- деякі моделі представлення можуть бути складнішими, ніж ті, які зберігаються безпосередньо в базі;
- є сенс робити перевірки бізнес-логіки на рівні об'єктів, які бачить користувач.

Відображення – представляє собою інтерфейс користувача. Може бути виконаний практично у будь-якому вигляді. Слугує для відображення та маніпуляції даними з боку користувача. По суті – проміжний рівень між моделлю та контроллером.

Контроллер – надає засоби для використання даних. Він відповідає за всі дії над моделями, перетворення їх з доменних у моделі передачі та у зворотному напрямі. Також контроллер обробляє події, які надсилає відображення і реагує на них.

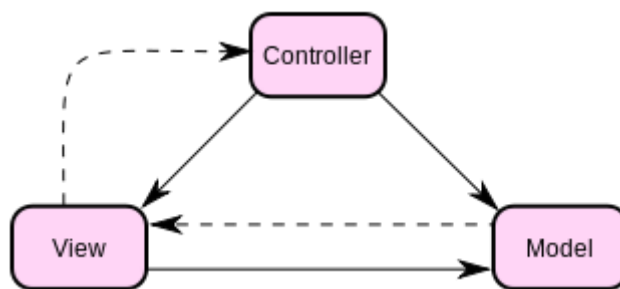


Рисунок 2.3 - Взаємодія компонентів шаблону MVC

2.2.2 MVVM

Патерн MVVM (Model-View-ViewModel) архітектурний патерн, який дозволяє розмежувати дані від відображення, як і MVC.

Даний патерн був представлений Джоном Госсманом в 2005 році як модифікація шаблону Presentation Model спеціально для застосування у технології Windows Presentation Foundation. Наразі цей патерн є широковідомим і застосовується для архітектури застосунків в найрізноманітніших технологіях, в тому числі при розробці під Android, iOS [13].

MVVM складається з трьох компонентів: моделі (Model), моделі подання (ViewModel) і представлення (View), що схоже на поділ моделей у шаблоні MVC .

Основна відмінність MVVM від MVC полягає в тому, що, на відміну від MVC, MVVM має один додатковий шар моделі, який називається ViewModel [14].

Це дозволяє застосовувати його в технологіях, де існує концепція зв'язування даних (Binding). У класичному MVC всі виклики до даних та їх зміна проходять через контроллер, який керує змінами. Тобто, інтерфейс ніяк не пов'язаний зі станом даних. Тому патерн MVC не дуже підходить для застосування в таких технологіях, оскільки концепція зв'язування даних не вкладається в рамки MVC.

Проте, патерн MVVM завоював досить широку популярність серед розробників, і є стандартним архітектурним рішенням для застосунків, написаних на платформі Андроїд.

Оскільки було вирішено використовувати технологію Xamarin для розробки, де використовується концепція зв'язування, патерн MVVM є кращим рішенням архітектурних задач.

2.3 Опис засобів проектування

Сучасний цикл розробки програмних систем важко уявити без етапу проектування. Дійсно, якщо система спроектована належним чином, це дозволяє побачити «вузькі місця» програмного продукту ще до початку розробки. Крім того, це забезпечує меншу кількість змін на етапі розробки, що, у свою чергу, дозволяє зекономити час та ресурси. Звісно, неможливо передбачити всі проблеми в умовах сучасного стрімкого розвитку технологій та змін ринку. Тому зміни вимог до програмного продукту під час безпосередньої розробки не уникнути. Але проектування значно економить ресурси і в таких умовах, оскільки дозволяє описувати програмні системи у зручний спосіб, описуючи як і різні структурні елементи, так і різні варіанти їх використання.

До того ж, етап проектування дозволяє створювати гарно документовані системи, що забезпечує швидше залучення нових людей до розробки.

2.3.1 CASE – системи

Найширше застосування на ринку розробки програмного забезпечення набули CASE-системи (з англ. computer-aided software engineering), які дозволяють автоматизувати більшість сучасних методів та підходів до проектування програмних продуктів [15].

Більшість програмних продуктів даної категорії надають користувачам наступні можливості:

- проектування багаторівневих систем різної складності;
- підтримка стандартного набору діаграм мови UML останніх версій;
- автоматизація розробки, що включає в себе генерацію коду мовами високого рівню на основі структурних схем та діаграм;
- автоматична генерація документації до програмної системи.

Таким чином, основними перевагами використання CASE-систем виступають:

- економія часу розробки застосунків;
- підвищення ефективності роботи програмістів за рахунок зменшення рутинної роботи і, як наслідок, зменшення кількості помилок і часу на тестування;
- підтримка проєктів різної складності;
- підтримка повторного використання розроблених структурних елементів;
- спрощення взаємодії учасників розробки.

Додатковою перевагою програмні системи такого типу можна назвати орієнтованість на бізнес. Дійсно, за рахунок можливостей генерації діаграм загальної структури систем, варіантів використання, бізнес- процесів, полегшується процес презентації цих систем для замовника, що є досить важливим етапом циклу розробки.

Проте, системи мають декілька суттєвих недоліків, серед яких можна назвати:

- висока вартість CASE-систем;
- необхідність додаткових категорій персоналу, який займається обслуговуванням систем;
- відсутність необхідності їх використання для невеликих проєктів.

2.3.2 Мова UML

Однією з переваг CASE-систем було названо підтримку мови UML. UML (від англ. Unified Modeling Language) – мова моделювання, яка використовується для опису програмних систем. Є стандартним засобом для опису програмних систем на різних рівнях, від абстрактної моделі систем до опису ієрархій класів у реалізації.

Мова застосовується на всіх етапах циклу розробки програмного забезпечення. Оскільки семантика мови має засоби для опису систем з різних точок зору – вона вважається універсальним засобом для опису програмного забезпечення і спрощує процес спілкування між групами розробників а також між представниками розробників та бізнесу. Окрім опису безпосередньо системи, UML має підтримку діаграм, які описують поведінку системи в різних ситуаціях. Такі діаграми включають в себе діаграми 3 типів:

- діаграми діяльності;
- діаграми станів;
- діаграми прецедентів.

Діаграма діяльності у вигляді блок-схеми описує граф діяльності, тобто послідовність дій, які ведуть від якогось вхідного сигналу до вихідного. Така діаграма спрощує розуміння певних процесів, що відбуваються в системі.

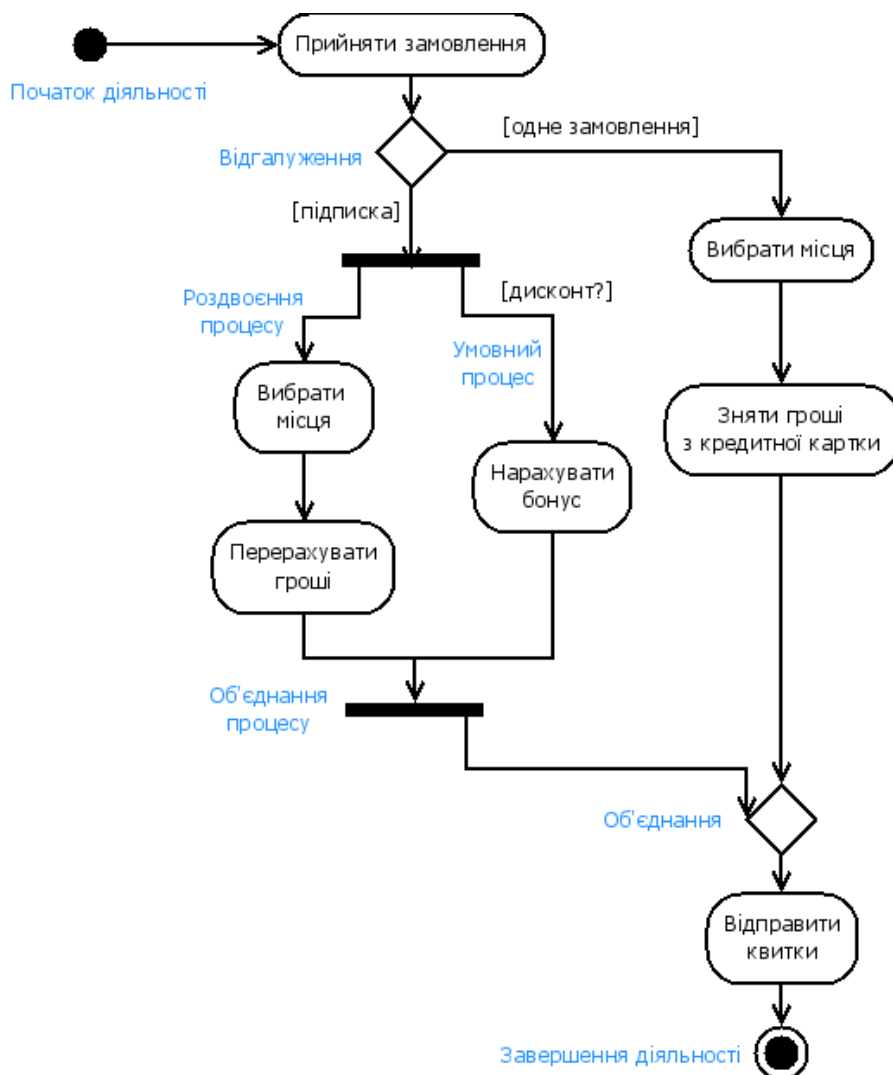


Рисунок 2.4 - Приклад діаграми діяльності

Діаграма станів, у свою чергу, описує, як зрозуміло за назвою, стани об'єктів та переходи зі стану у стан. Спрощує розуміння циклу життя об'єктів, оскільки показує множину його станів а також дії, які призводять до його зміни.

Діаграма прецедентів або діаграма варіантів використання показує взаємодію між акторами та прецедентами в системі. Головна мета діаграми такого типу – показати, які послуги система надає різним сутностям, які називають акторами. Зазвичай актори – користувачі системи різного роду. Оскільки окрім безпосередньо користувачів у системі можуть бути залучені інші люди (наприклад, адміністратор, продавець, тощо), діаграма наочно

показує, які категорії користувачів мають бути залучені у певних процесах. При цьому, діаграма не показує, як мають бути реалізовані взаємодія.

Дуже важливою структурною діаграмою мови UML є діаграма класів. Діаграма класів описує класи, типи даних та їх відношення. Зазвичай застосовується для об'єктно-орієнтованих мов програмування.

Великою перевагою мови UML є її наочність а також велика кількість програмних засобів для їх створення, що розповсюджуються вільно. Також, багато середовищ для розробки програмного забезпечення мають інтегровані можливості створення діаграм, але зазвичай, з обмеженою функціональністю (тільки діаграми класів).

Також мова має декілька недоліків, серед яких основними є надмірність мови та неточність семантики. Незважаючи на існування стандарту, інтерпретація діаграм може різнитись від прокту до проєкту. До того ж необхідно ввести правила цієї інтерпретації.

2.4 Висновки до розділу 2

Отже, в даному розділі було описано основні підходи та засоби розробки і проєктування програмних застосунків під мобільні платформи, зокрема, платформу Android. В якості технології розробки було обрано Xamarin, оскільки вона дозволяє використовувати більшість можливостей обраної платформи а також надає можливість використовувати спільні інтерфейси для різних платформ. Відповідно, в якості основної мови було обрану мову C#, яка є основною для цієї технології. В якості засобу проєктування було обрано вбудовані можливості середовища Visual Studio.

РОЗДІЛ 3. ОСОБЛИВОСТІ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОБІЛЬНОГО ДОДАТКУ

3.1 Опис модулів ПЗ

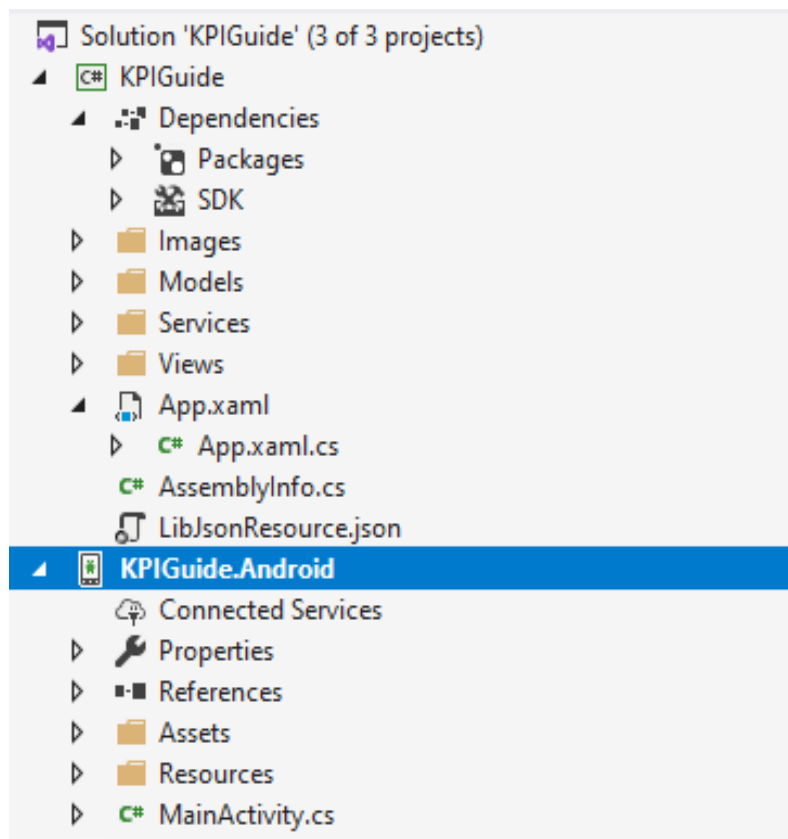


Рисунок 3.1 - Загальна структура застосунку

Загальна структура розробленого застосунку наведена на рисунку 3.1. Проект з назвою KPIGuide відповідає за функціональність, яка може бути використана для додатків на різних платформах, і представляє собою бібліотеку, розроблену під стандарт .NET Standart 2.0.

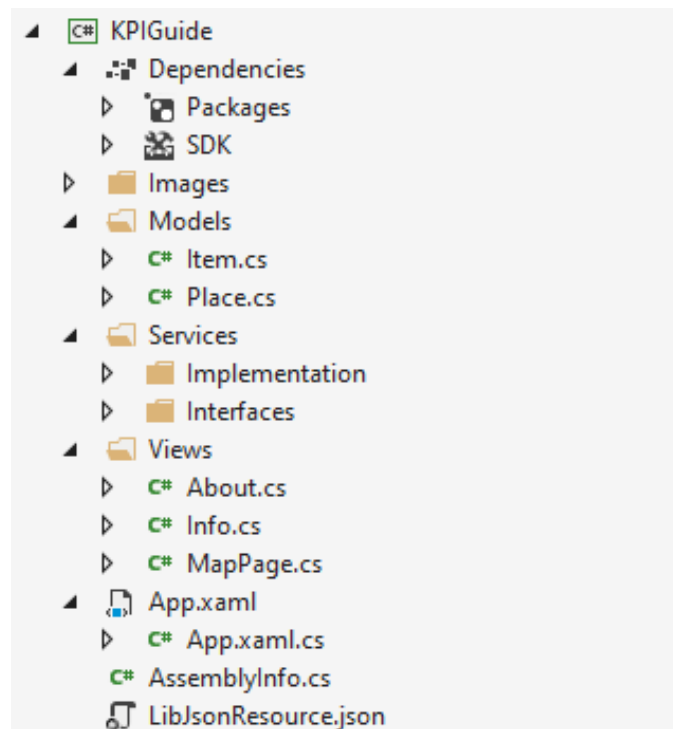


Рисунок 3.2 - Структура проєкту KPIGuide

На рисунку 3.2 показано структуру проєкту KPIGuide.

Основні сервіси знаходяться в папці Services. У папці Interfaces знаходяться інтерфейси, в папці Implementation – реалізація сервісів. Така структура розроблена для можливості реалізації інтерфейсів під різні платформи. Оскільки була обрана платформа Android в якості цільової, було вирішено помістити реалізацію інтерфейсів тут для зменшення витрат на реєстрацію інтерфейсів в контейнері ін'єкції залежності (з англ. Dependency injection container). Ця операція використовує багато ресурсів, оскільки для використання використовується механізм рефлексії, який вимагає аналізу метаданих.

Папка Views призначена для класів, які використовуються для імплементації інтерфейсу. Було вирішено відмовитись від патерну MVVM, оскільки немає гострої необхідності у структуризації проєкту і це тільки ускладнить проєкт. Також було вирішено виконати інтерфейс у вигляді коду на C#, а не у вигляді XAML розмітки, щоб динамічно проініціалізувати початкові

дані одразу при створенні. Також необхідно було пов'язати сервіси зі сторони, які вони потребують. В якості параметрів було використано інтерфейси відповідних сервісів для того, щоб у разі зміни імпліmentaції не виникало змін у інших класах. Також це дозволить, за необхідності, змінити спосіб зв'язування на ін'єкцію залежності, наприклад, при реалізації для іншої платформи.

3.2 Опис механізму сховища даних

Технологія Xamarin підтримує 3 основні механізми збереження даних:

- файлова система;
- локальні бази даних;
- Azure сервіси.

Було вирішено відмовитись від локальної бази даних або хмарної бази даних на Azure на даному етапі. Наразі даних не настільки багато, вони займають не більше 10 МБ простору, а структура даних легко вміщується в один об'єкт. Тому було вирішено використати файлову систему. Єдиний недолік такого підходу у різних реалізаціях файлових систем на різних платформах.

Для усунення цього недоліку було використано механізм вбудованих файлів у проєкт бібліотеки. Це дозволить зменшити зв'язність коду з конкретною платформою, що може бути корисним при перенесенні на інші платформи.

Однак, оскільки вміст спільного проєкту збирається у залежні від нього проєкти (у даному випадку мається на увазі проєкти під певну платформу), префікс, що використовується для вбудованих ідентифікаторів файлових ресурсів, може змінюватися. Це означає, що ідентифікатор ресурсу для кожного вбудованого файлу може бути різним для кожної платформи. Це треба враховувати при перенесенні ресурсів на інші платформи.

Дані про кожну локацію описані у класі Place. Код класу Place поміщена на лістинг 1.

Лістинг 1. Клас Place

```
using System;
using System.Collections.Generic;
using System.Text;

namespace KPISystem.Models
{
    public class Place
    {
        public string Name { get; set; }
        public string Description { get; set; }
        public List<string> ImagePath { get; set; }
        public List<string> Addresses { get; set; }
    }
}
```

Для збереження цих даних було вирішено використати формат JSON, який досить економний щодо місця, до того ж існує багато бібліотек для його десеріалізації. Ще однією перевагою цього формату є його платформонезалежність.

Кожен об'єкт локації має набір фотографій. Для спрощення проєкту в кожен об'єкт входить перелік шляхів для всіх фотографій, які теж вбудовані в проєкт.

У випадку вбудованого ресурсу ідентифікатор ресурсу має форму Namespace.Filename.Extension. Можна впорядкувати вбудовані ресурси в папки для структуризації. Коли вбудований ресурс розміщується у папці, ім'я папки стає частиною ідентифікатора ресурсу (розділеного періодами), таким чином формат ідентифікатора ресурсу стає Namespace.Folder.Filename.Extension.

Всі фотографії розміщено в папці Image і розбито по вкладених папках, які мають назву K і номер корпусу (рисунок 3.4)

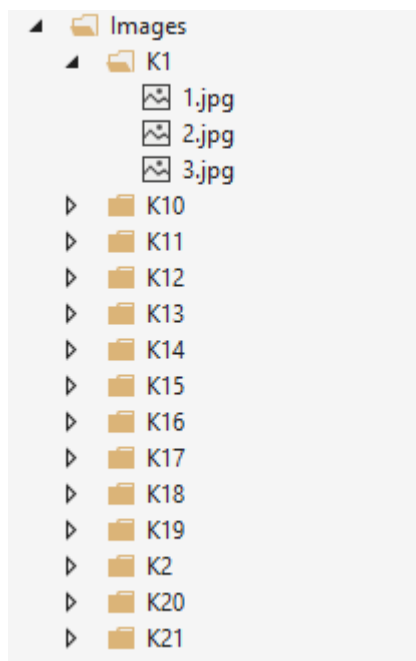


Рисунок 3.3 - Структура каталогів для фотографій

3.3 Опис використаних компонентів

Для відображення карти використовується елемент `Xamarin.Forms.Maps`. Він використовує стандартну реалізацію карт для певної платформи (у випадку платформи Android – Google Maps, у випадку iOS – Apple Maps). Специфікація API для цих карт не дозволяє оффлайн використання, тож ніяким чином не можна мати до них доступ без підключення до мережі Інтернет, окрім як малюванням власної карти.

Бібліотеку `Xamarin.Forms.Maps` необхідно ініціалізувати різним чином на різних платформах, оскільки він вимагає отримання дозволів на отримання геоданих, механізм яких різниться в залежності від цільової платформи.

Опишемо процес ініціалізації для платформи Android. Процес включає 6 кроків:

- Отримати ключ API Google Maps і додати його до маніфесту Android.
- Вказати номер маніфесту служб Google Play.
- Вказати вимогу для бібліотеки Legacy Apache HTTP Legacy у маніфесті.

- Вказати дозвіл WRITE_EXTERNAL_STORAGE у маніфесті.
- Вказати дозволи на розташування в маніфесті.
- Запит дозволу на розташування для виконання в класі MainActivity.

Приклад маніфесту для проєкту вказаний на лістингу 2.

Лістинг 2. Маніфест додатку Android

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1" android:versionName="1.0"
    package="com.companyname.workingwithMaps">
    <uses-sdk android:minSdkVersion="21" />
    <application android:label="WorkingWithMaps.Android">
        <meta-data android:name="com.google.android.geo.API_KEY"
            android:value="INSERT-YOUR-API-KEY-HERE" />
        <meta-data android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
        <!-- Necessary for apps that target Android 9.0 or higher -->
        <uses-library android:name="org.apache.http.legacy"
            android:required="false" />
    </application>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
    />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
    />
</manifest>
```

Щоб використовувати функцію "Google Maps" в Android, необхідно зареєструвати ключ API Maps в Google. Без виконання цього кроку на екрані відображається лише сітка, без самої карти. Потрібно отримати ключ Android API Android Maps v2 - ключі зі старої версії API Google Maps v1 не працюватимуть.

Отримання ключа API v2 включає наступні кроки:

- необхідно отримати відбиток SHA-1 від сховища, який використовується для підписання програми;
- створити проєкт у консолі API Google;

- отримати ключ API.

Докладна інструкція для цього кроку є на офіційній сторінці Google Maps API.

У випадку, якщо додаток Xamarin.Forms має цільовий API 28 або новішої версії, необхідно додати наступну декларацію до елемента `<application>` `AndroidManifest.xml`:

```
<uses-library android:name="org.apache.http.legacy" android:required="false" />
```

Ця директива дозволяє програмі використовувати клієнтську бібліотеку Apache Http, яку було видалено з кореневого елемента завантажувального пакету програми в Android 9.

Вказати дозвіл `WRITE_EXTERNAL_STORAGE` у маніфесті необхідно лише у випадку використання API версій 22 і нижче. Оскільки ми використовуємо версію 28, цей крок не є необхідним.

Оскільки застосунок, що розроблюється, має отримати доступ до місцезнаходження користувача, потрібно надіслати запит на дозвіл, додавши до маніфесту дозвіл `ACCESS_COARSE_LOCATION` або `ACCESS_FINE_LOCATION` як дочірнього елемента маніфесту, як вказано на лістингу 3.

Лістинг 3. Дозвіл на запит до геоданих пристрою

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  android:versionCode="1" android:versionName="1.0"
  package="com.companyname.myapp">
  ...
  <uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
  />
</manifest>
```

Дозвіл `ACCESS_COARSE_LOCATION` дозволяє API використовувати Wi-Fi або мобільні дані або те і інше для визначення місця розташування пристрою.

Дозвіл ACCESS_COARSE_LOCATION дозволяє API використовувати Wi-Fi або мобільні дані або те і інше для визначення місця розташування пристрою.

Дозволи ACCESS_FINE_LOCATION дозволяють API використовувати (GPS), Wi-Fi або мобільні дані, щоб визначити точне місце розташування.

Оскільки застосунок використовує API 28 як цільовий та потребує доступу до місцезнаходження користувача, необхідно перевірити, чи є у нього потрібний дозвіл під час виконання, і запитувати його, якщо його немає. Це можна здійснити командами, які вказані на лістингу 4.

Лістинг 4. Дозвіл на запит до геоданих пристрою під час виконання

```
const int RequestLocationId = 0;

readonly string[] LocationPermissions =
{
    Manifest.Permission.AccessCoarseLocation,
    Manifest.Permission.AccessFineLocation
};
```

Також треба додати до класу MainActivity (клас основного вікна додатку у платформі Android код, який вказано на лістингу 5.

Лістинг 5. Клас MainActivity

```
protected override void OnStart()
{
    base.OnStart();

    if ((int)Build.VERSION.SdkInt >= 23)
    {
        if CheckSelfPermission(Manifest.Permission.AccessFineLocation) !=
            Permission.Granted)
        {
            RequestPermissions(LocationPermissions,
                RequestLocationId);
        }
        else
        {
            // Permissions already granted - display a message.
        }
    }
}
```

За умови, що програма націлена на API 23 або вище, цей код виконує перевірку дозволу виконання часу на дозвіл AccessFineLocation. Якщо дозвіл не

надано, запит на дозвіл робиться за допомогою виклику методу RequestPermissions. Для цього треба перевизначити метод OnRequestPermissionsResult в класі MainActivity. Приклад цього вказано на лістингу 6.

Лістинг 6. Метод OnRequestPermissionsResult з класу MainActivity

```
public override void OnRequestPermissionsResult(int requestCode, string[]
permissions, [GeneratedEnum] Permission[] grantResults)
{
    if (requestCode == RequestLocationId)
    {
        if ((grantResults.Length == 1) && (grantResults[0] ==
(int)Permission.Granted))
            // Permissions granted - display a message.
        else
            // Permissions denied - display a message.
        }
    else
    {
        base.OnRequestPermissionsResult(requestCode, permissions,
grantResults);
    }
}
```

Функціональність компоненту Map дещо обмежена в порівнянні з відповідними додатками для карт. Причини цього полягають у тому, що компонент використовує API для карт, специфікація яких обмежена. До того ж, компонент надає однаковий інтерфейс для карт всіх платформ, і має реалізувати тільки однакові можливості.

Клас Map визначає такі властивості, які керують зовнішнім виглядом та поведінкою карти:

- IsShowingUser типу bool вказує, чи карта показує поточне місцезнаходження користувача;
- ItemSource, типу IEnumerable, який визначає колекцію IEnumerable елементів, які підлягають відображенню;
- ItemTemplate типу DataTemplate, який визначає шаблон DataTemplate, який слід застосувати до кожного елемента колекції відображених елементів;

- ItemTemplateSelector типу DataTemplateSelector, який визначає DataTemplateSelector, який буде використовуватися для вибору шаблону DataTemplate для елемента під час виконання застосунку;
- HasScrollEnabled типу bool визначає, чи дозволено прокручувати карту;
- HasZoomEnabled, тип bool, визначає, чи дозволяється масштабувати карту;
- MapElements типу IList <MapElement> представляє список елементів на карті, таких як полігони та полілінії;
- Type типу MapType вказує стиль відображення карти;
- MoveToLastRegionOnLayoutChange типу bool контролює, чи відображатиметься відображена область карти зі своєї поточної області до раніше встановленої області, коли відбувається зміна макета;
- Pins типу IList <Pin> представляють список міток на карті;
- VisibleRegion типу Mapspan повертає область на карті, яка відображається зараз.

Ці властивості, за винятком властивостей MapElements, Pins та VisibleRegion, підтримуються об'єктами BindableProperty, що означає, що вони можуть бути цілями прив'язки даних.

Клас Map також визначає подію MapClicked, яка запускається при натисканні на карту. Об'єкт MapClickedEventArgs, який є аргументом події і містить інформацію про неї, має єдину властивість під назвою Position. Коли подія запускається, властивість Position відображає місце на карті, яке було натиснуто.

Як було вказано вище, клас Map має властивість MapType, яка вказує, який тип карти відображається. Властивість Map.MapType є типом перелічення MapType, щоб визначити стиль відображення карти. Перерахування MapType визначає наступні члени:

- Street:
- Satellite:

– Hybrid.

Street означає, що буде показана звичайна карта (за замовчуванням).
Satellite означає, що буде показана карта із супутниковими зображеннями.
Hybrid означає, що буде показано супутникове зображення, з накладеною поверх нього картою (рисунк 3.5).

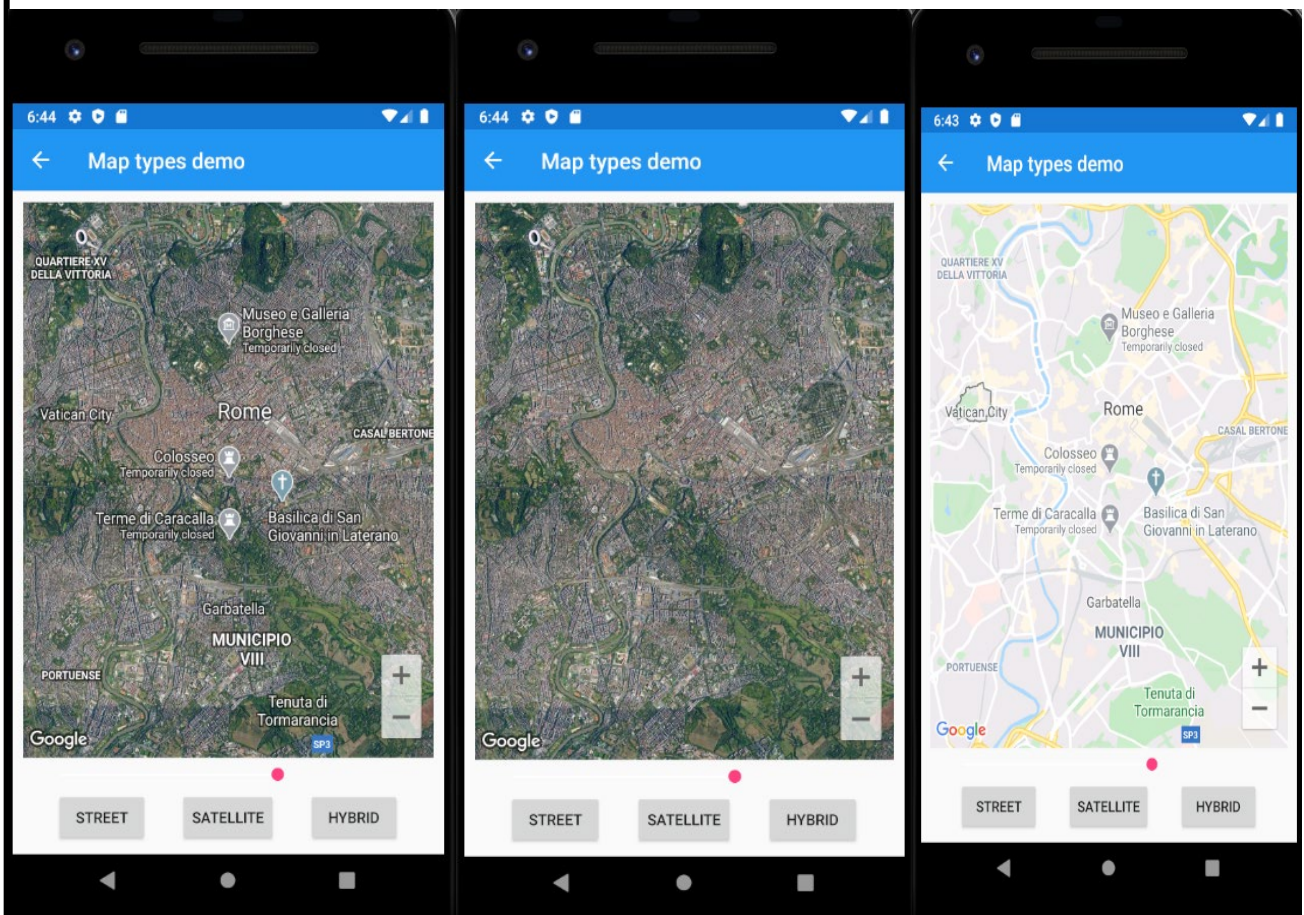


Рисунок 3.5 - Зображення 3 типів об'єкту Map

Поведінку карти можна перевизначити за допомогою обробника OnClick. У застосунку це буде використано в одному з режимів, коли замість визначення напряму можна буде натиснути на карту і відобразити інформацію про об'єкт, на який було натиснуто.

Також бібліотека надає клас Geocoder. Цей клас має 2 методи :

– GetAddressesForPositionAsync(Position);

– `GetPositionsForAddressAsync(String)`.

Метод `GetAddressesForPositionAsync(Position)` приймає у якості параметру об'єкт `Position`, і повертає масив можливих адрес для нього (деякі об'єкти мають декілька адрес). Метод `GetPositionsForAddressAsync(String)` приймає у якості параметру об'єкт `String`, і повертає масив можливих точок (об'єктів `Position`).

Для визначення напрямку телефону необхідно використати компас. Android не надає API для отримання заголовка компаса. Google рекомендує використовувати акселерометр та магнітометр для обчислення напрямку на північ. Для доступу до таких сенсорів необхідно підключити бібліотеку `Xamarin.Essentials`.

У рідкісних випадках можуть бути непослідовні результати, оскільки датчики потрібно відкалібрувати, що передбачає переміщення пристрою в русі цифрою-8. Найкращий спосіб зробити це - відкрити Карти Google, торкнутись крапки для поточного положення місця та обрати калібрувальний компас.

Зважаючи на те, як оновлюються та обчислюються значення Android компаса, виникла потреба у вирівнюванні значень. Можна застосувати фільтр низьких частот, який в середньому визначає значення синуса і косинуса кутів, і ввімкнути його за допомогою перевантаження методу `Start`, який приймає параметр `bool applyLowPassFilter` у якості другого параметру.

3.4 Особливості реалізації сервісу для розрахунку напрямку

Ідея, яка полягає у реалізації сервісу розрахунку напрямку, полягає у вирахунку кута відхилення телефону від півночі.

Оскільки платформа Android не надає API для компаса, було вирішено взяти API для сенсорів, а саме акселерометру та магнітометру. Google радить використовувати саме цей підхід. Однією з додаткових переваг використання

саме такого підходу є те, що доступ до сенсорів не вимагає додаткових дозволів від користувача, що спрощує реалізацію.

Також було вирішено зробити свою реалізацію інтерфейсів компасу, який надає бібліотека Xamarin.Essentials. Такий вибір було зроблено для спрощення реалізації та приховування деяких деталей обрахунку величин. Код методів ініціалізації та оброблення події зміни показників вищезгаданих сенсорів наведено на лістингах 7 та 8 відповідно.

Лістинг 7. Ініціалізація компасу

```
public class Compass: Java.Lang.Object, ICompass, IDisposable,
    ISensorEventListener
{
    private SensorManager _sensorManager;
    private readonly Sensor _accelerometer;
    private readonly Sensor magnetometer;
    private List<float> _lastAccelerometer = new List<float>();
    private List<float> _lastMagnetometer = new List<float>();
    private bool lastAccelerometerSet;
    private bool lastMagnetometerSet;
    private List<float> _r = new List<float>();
    private List<float> _orientation = new List<float>();

    private bool listening;

    void Init()
    {
        var ctx = Application.Context;
        if (ctx == null)
        {
            System.Diagnostics.Debug.WriteLine("Context not found,
            can not start.");
            return;
        }
        if (_sensorManager == null)
            _sensorManager =
            ctx.GetSystemService(Context.SensorService) as SensorManager;
```

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

51

```

        if (_accelerometer == null)
            _accelerometer =
sensorManager?.GetDefaultSensor(SensorType.Accelerometer);

        if (_magnetometer == null)
            _magnetometer =
sensorManager?.GetDefaultSensor(SensorType.MagneticField);
    }

```

Лістинг 8. Обробник події CompassChanged

```

protected virtual void OnCompassChanged(CompassChangedEventArgs e) =>
    CompassChanged?.Invoke(this, e);
/// <summary>
///
/// </summary>
/// <param name="e"></param>
public void OnSensorChanged(SensorEvent e)
{
    lock(locker)
    {
        if (e.Sensor == _accelerometer && !_lastAccelerometerSet)
        {
            CopyValues (e.Values, _lastAccelerometer);
            _lastAccelerometerSet = true;
        }
        else if (e.Sensor == _magnetometer &&
!_lastMagnetometerSet)
        {
            CopyValues (e.Values, _lastMagnetometer);
            _lastMagnetometerSet = true; }
        if (_lastAccelerometerSet && _lastMagnetometerSet)
        {
            SensorManager.GetRotationMatrix(r, null,
_lastAccelerometer, _lastMagnetometer);
            SensorManager.GetOrientation(_r, _orientation);
            var azimuthInRadians = _orientation.First();
            var azimuthInDegrress =
(Java.Lang.Math.ToDegrees(azimuthInRadians) + 360.0) % 360.0;

            OnCompassChanged(new
CompassChangedEventArgs(azimuthInDegrress));

```

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

52

```

        _lastMagnetometerSet = false;
        _lastAccelerometerSet = false;
    }
}

```

Було розроблено власний компонент для платформи Android, який сумісний з бібліотекою Xamarin.Forms. За необхідністю також можна розробити такий компонент для інших платформ для портування застосунку. Загальну ідею створення такого роду компонентів було з репозиторію подібного компоненту на сайті GitHub [17].

3.5 Особливості реалізації сервісу розрахунку зміщення у певному напрямі

Сервіс розрахунку зміщення у певному напрямі використовує 2 сервіси – сервіс компасу та сервіс визначення місцезнаходження, який надається компонентом Map з бібліотеки Xamarin.Forms.Maps. Принцип його роботи заснований на обрахунку різниці між поточним положенням та положенням наступної точки на азимуті напрямку телефону з певним кроком. За замовчуванням крок рівний 20 метрам.

При цьому обрахунку постала проблема обрахунку різниці між поточною точкою та наступною при зміщенні, оскільки позиція задається географічними координатами, тобто широтою та довготою.

Широта – це координата у сферичній системі координат, яка позначає відхилення точки від екватору. Позначається у градусах від -90° до $+90^\circ$. Тобто, широта точки на екваторі - 0° , широта південного полюсу -90° , а широта північного, відповідно, $+90^\circ$.

Довгота – координата у сферичній системі координат, яка позначає відхилення точки від Гринвіцького меридіану. Меридіан – уявна лінія, яка з'єднує полюси Землі. Вимірюється від -180° до $+180^\circ$. В Україні історично склалося так, що довгота вимірювалась від 0 до 360° на схід, але оскільки при використанні компаса у платформі Android виміри ідуть від -180° до 180° , було

вирішено використовувати відповідні позначення. На території України така зміна несуттєва, оскільки на схід довгота вважається додатною величиною, а значення не перевищують 180°.

Оскільки область роботи додатку – тільки територія університету, можна знехтувати вигином Землі і вважати, що робота іде з декартовими координатами на площині. Це суттєво спрощує задачу, оскільки можна вважати, що точка поточного місцезнаходження пристрою – нульова. Тоді можна вважати, що широта – це координата Y, а довгота – координата X. При такому допущенні можна розрахувати зміну відповідних координат у залежності від азимуту напрямку телефону, використовуючи одиничне коло і значення тригонометричних функцій на ньому. Після розрахунку функцій необхідно їх значення помножити на довжину кроку, яка за замовчуванням рівна 20 метрам. Прийmemo кут, який повертає компас, за x , а крок за 1. Також необхідно зазначити, що кут повертається у градусах, а тригонометричні функції у мові C# працюють з радіанами, тому у формулі треба врахувати приведення до радіанів. Тоді отримаємо формули для зміни довготи і широти відповідно:

$$\Delta Y = \sin\left(\frac{\pi}{180} * x\right) * l \quad (4.1)$$

$$\Delta X = \cos\left(\frac{\pi}{180} * x\right) * l \quad (4.2)$$

Формули 4.1 і 4.2 використовуються в сервісі обрахунку позиції найближчої локації, код якого поміщено на лістинг 8.

Лістинг 8. Клас PositionCalculatorService

```
using KPIGuide.Services.Interfaces;
using static System.Math;
namespace KPIGuide.Services.Implementation
{
    public class PositionCalculatorService :
    IPositionCalculatorService
    {
        const double _delta = 0.000150;
        public (double Latitude, double Longitude)
        GetNewPosition((double Latitude, double Longitude) position, double
        angle)
    }
}
```

```

        {
            var deltaX = Cos((PI / 180) * angle);
            var deltaY = Sin((PI / 180) * angle);
            var result = (Latitude: position.Latitude + deltaY *
                _delta,
                Longitude: position.Longitude + deltaX * _delta);
            return result;
        }
    }
}

```

3.6 Особливості реалізації допоміжних сервісів

Використовуючи сервіс, описаний в попередньому підрозділі, можна знайти найближчу локацію на певній відстані, яка знаходиться в бік напрямку телефону. Для цього реалізовано цикл в головному вікні, який використовує всі розроблені у застосунку сервіси.

Наведемо перелік допоміжних сервісів:

- GeocoderService;
- PlacesDataStore;
- PositionCalculatorService;
- ReadService.

GeocoderService – обгортка над об'єктом Geocoder. Надає подібний функціонал, з єдиною відмінністю. Він повертає не перелік адрес за вказаною точкою, а одну адресу, яка підходить під вимоги застосунку. Зазвичай кожен корпус має адресу, яка містить позначку корпус. Ця адреса і буде повертатись, якщо вона існує.

PlacesDataStore – сервіс, який слугує сховищем для інформації про корпуси. Під час старту програми, вся інформація про корпуси завантажується у пам'ять для пришвидшення доступу до неї. Описаний механізм є актуальним для невеликих наборів даних. Оскільки дані мають невеликий обсяг, було вирішено застосувати його у цій ситуації. Необхідно зазначити, що фотографії не завантажуються одразу, оскільки об'єкти Place мають лише шлях до них.

Більше того, оскільки кількість фотографій заздалегідь невідома, завантаження їх не є доцільним з точки зору динамічної побудови візуального інтерфейсу, незважаючи на їх невеликий загальний обсяг.

Окрім збереження, сервіс PlacesDataStore надає інтерфейс для маніпулювання даними. На лістингу 9 наведено код інтерфейсу, що надається.

Лістинг 9. Інтерфейс IDataStore

```
using System.Collections.Generic;
using System.Threading.Tasks;

namespace KPIGuide.Services.Interfaces
{
    public interface IDataStore<T>
    {
        bool AddItem(T item);
        bool UpdateItem(T item);
        bool DeleteItem(string id);
        T GetItem(string id);
        IEnumerable<T> GetItems(bool forceRefresh = false);
    }
}
```

Окрім вищезазначених методів, клас PlacesDataStore надає методи GetAllAddresses та GetPlaceByAddress. Ці методи розроблені для того, щоб можна було визначити, чи належить певна адреса якомусь з корпусів. Також метод GetPlaceByAddress корисний у другому режимі, який дозволяє обрати місце, про яке необхідно вивести інформацію. Тоді після визначення адреси можна одразу отримати інформацію про певний корпус.

ReadService – сервіс, який запускається під час старту додатку. Він зчитує всі дані про корпуси з файлу LibJsonResource.json, і надає їх PlacesDataStore об'єкту для збереження.

Всі ці сервіси необхідні для визначення найближчої локації. Для цього в циклі запускається сервіс PositionCalculatorService, який рахує зміщення позиції на певну відстань. Далі GeocoderService повертає адресу отриманої точки. Після цього PlacesDataStore повертає всі адреси і виконується перевірка належності знайденої адреси до переліку адрес всіх локацій. Якщо вона належить цій множині, тоді запускається вікно виводу інформації про локацію, куди

передається знайдений за адресою об'єкт. Якщо ж адреса не належить множині всіх адрес, тоді цикл продовжується.

Також існує ймовірність, що телефон буде спрямовано у сторону, де немає будівель. Щоб зекономити час на розрахунки, необхідно обмежити зміни координат певними значеннями. Було вирішено встановити максимальний крок у 100 метрів. Також було вираховано максимальні і мінімальні значення географічних координат, у межах яких ведеться пошук. Оскільки корпуси університету розташовані не тільки на території кампусу, було вирішено взяти координати у наступному проміжку: від [49° N, 29° E] до [51° N, 31°E]. Таких обмежень більше ніж достатньо.

3.7 Рекомендації щодо подальшого вдосконалення

Подальшими напрямками розвитку програмної системи можна вважати наступні:

- розширення території охоплення застосунку до межі району або міста Києва;
- перенесення даних до хмарної бази даних, наприклад, Azure CosmosDB;
- портування застосунку на інші платформи (iOS, UWP);
- додавання можливості створювати свої маршрути;
- додавання можливості оновлювати дані шляхом викладання нових фото, описувати неописані будівлі;
- додавання можливості відтворювати мультимедійні файли, наприклад, відео;
- інтеграція розташування з картами Google (поділитись локацією в Google Maps);
- можливість обмінюватись маршрутами.

Варто зазначити, що завдяки обраній технології (Xamarin) портування на інші платформи і використання певних хмарних сервісів суттєво спрощене. Також у майбутньому на основі цього додатку можна створити стартап з монетизацією і розповсюджувати її за допомогою Google Play та App Store.

3.8 Висновки до розділу 3

У даному розділі було описано загальну структуру розробленого застосунку. Також було наведено можливості компонентів бібліотек Xamarin.Forms та Xamarin.Essentials, які було використано. Було описано засоби, використані для збереження даних про локації університету та структуру цих даних. описано шляхи реалізації основних та допоміжних сервісів застосунку. Було описано метод знаходження найближчої локації, наведено формули, які використовують для розрахунку, було описано реалізація цього методу. Також було описано можливі шляхи подальшого вдосконалення розробленого застосунку.

					ІАЛЦ.045490.004 ПЗ	Арк.
						58
Змін.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Метою даного дипломного проєкту була розробка програмного забезпечення «Додаток на базі Android "Мобільний гід по університету"».

На основі аналізу альтернативних рішень було визначено, що система має мати можливість визначати найближчу до користувача локацію, яка розташована в напрямі, куди спрямовано телефон. Аналіз засобів реалізації застосунків на платформі Android було показано доцільність створення такого програмного забезпечення на платформі .NET Core з використанням бібліотек Xamarin.Forms та Xamarin.Essentials. Відповідно до вимог платформи було обрано мову C# в якості основної.

Розроблена програмна система:

- надає можливість знайти поточне місцезнаходження;
- дозволяє знайти найближчу до користувача локацію в напрямі спрямування телефону;
- дозволяє вивести інформацію про локацію;
- надає можливість обрати локацію на карті;
- дозволяє подивитись загальну інформацію про університет.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Огляд платформи Windows Mobile. *Gamesmart* : URL:
http://gamesmart.at.ua/publ/obzor_mobilnykh_os_platform/obzor_platformy_windows_mobile/2-1-0-10 (дата звернення 30.12.2020)
2. Введение в разработку для платформы Android. *Android* : URL:
<http://android-club.com.ua/?articles=15> (дата звернення 30.12.2020)
3. Огляд платформи Symbian. *Gamesmart* : URL:
http://gamesmart.at.ua/publ/obzor_mobilnykh_os_platform/obzor_platformy_symbian/2-1-0-8 (дата звернення 30.12.2020)
4. Зdziarski Джонатан, iPhone Разработка приложений с открытым кодом - Санкт-Петербург: БХВ-Петербург, 2009, 386 стр. (дата звернення 16.01.2020)
5. Карти Google. *Google Карти* : URL:
<https://www.google.com/intl/ru/maps/about/> (дата звернення 26.01.2020)
6. Google Maps. *Google Maps* : URL:
https://www.uk.w3ki.com/google_maps/ (дата звернення 26.01.2020)
7. Карти Google. *Google Maps* : URL:
https://www.uk.w3ki.com/google_maps/index.html (дата звернення 16.02.2020)
8. Google Earth. *Google* : URL:
<https://www.google.com.ua/intl/ru/earth/> (дата звернення 16.02.2020)
9. Система об'єктно-орієнтованого програмування. *Nbuv*: URL:
http://nbuv.gov.ua/j-pdf/komp_2011_4_4.pdf (дата звернення 20.02.2020)
10. Основи програмування мовою Java. *Ami.lnu* : URL:
<https://ami.lnu.edu.ua/wp-content/uploads/2017/05/Java.pdf> (дата звернення 20.02.2020)
11. Мова програмування Kotlin. *Metanit*: URL:
<https://metanit.com/kotlin/tutorial/1.1.php> (дата звернення 20.02.2020)

12. Mvc. *SchoolboyProgrammer* : URL:

<https://schoolboyprog10.blogspot.com/p/mvc-mvc-model-view-controller.html> (дата звернення 26.03.2020)

13. Mvvm. *Edu_ua*: URL:

<http://krs.chmnu.edu.ua/bitstream/123456789/628/1/Автореферат Григорьев Д.О 403.pdf> (дата звернення 26.03.2020)

14. Паттерн MVVM. *Metanit*: URL:

<https://metanit.com/sharp/wpf/22.1.php> (дата звернення 26.04.2020)

15. Case-средства. *bourabai*: URL:

<http://bourabai.kz/einf/case.htm> (дата звернення 26.04.2020)

16. CompassPlugin. *Github* : URL:

<https://github.com/JarleySoft/CompassPlugin> (дата звернення 26.04.2020)